

ITS

Institut
Teknologi
Sepuluh Nopember

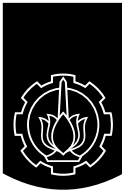
TUGAS AKHIR - TE 141599

**RANCANG BANGUN SISTEM PENGENALAN SIMBOL
TOILET DI DALAM RUANGAN DENGAN METODE SURF
UNTUK TUNA NETRA BERBASIS KAMERA**

Tonny Feriandi
NRP 2210100121

Dosen Pembimbing
Ronny Mardiyanto, S.T., M.T., Ph.D.
Dr. Ir. Djoko Purwanto, M.Eng.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

***INDOOR TOILET SYMBOL RECOGNITION SYSTEM USING
SURF ALGORITHM FOR THE VISUALLY IMPAIRED BASED
ON CAMERA***

Tonny Feriandi
NRP 2210100121

Advisor
Ronny Mardiyanto, S.T., M.T., Ph.D.
Dr. Ir. Djoko Purwanto, M.Eng.

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty Of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**RANCANG BANGUN SISTEM PENGENALAN
SIMBOL TOILET DI DALAM RUANGAN
DENGAN METODE SURF UNTUK TUNA NETRA
BERBASIS KAMERA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I



Renny Mardiyanto, S.T., M.T., Ph.D.
NIP. 198101182003121003

Dosen Pembimbing II



Dr. Ir. Dioko Purwanto, M.Eng.
NIP. 196512111990021002



KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

Rancang Bangun Sistem Pengenalan Simbol Toilet di Dalam Ruangan dengan Metode Surf untuk Tuna Netra berbasis Kamera

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Bapak, Ibu, kakak serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
2. Dr. Ronny Mardiyanto, ST., MT. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Dr. Ir. Djoko Purwanto, M.Eng. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Tasripan, IR. MT selaku Koordinator Bidang Studi Elektronika.
5. Dr. Tri Arief Sardjono, ST, MT selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
6. Seluruh dosen bidang studi elektronika.
7. Teman-teman laboratorium Elektronika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Akhirnya penulis berharap agar Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak serta pengembangan aplikasi *image processing*. Penulis juga berharap supaya Tugas Akhir ini dapat menjadi aplikasi yang lebih bermanfaat.

Surabaya, 21 Oktober 2014

Penulis



RANCANG BANGUN SISTEM PENGENALAN SIMBOL TOILET DI DALAM RUANGAN DENGAN METODE SURF UNTUK TUNA NETRA BERBASIS KAMERA

Nama : Tonny Feriandi
Pembimbing I : Ronny Mardiyanto, ST., MT., PhD
Pembimbing II : Dr.Ir Djoko Purwanto, M.Eng

ABSTRAK

Penginderaan visual merupakan suatu proses pengolahan data visual menjadi data referensi yang dijadikan dasar untuk menggambarkan objek visual. Proses tersebut dapat digunakan untuk membantu penyandang tuna netra mengenali ruang toilet. Pada dasarnya penyandang tuna netra tidak mampu mengenali lingkungan sekitarnya, sehingga dengan bantuan pengenalan simbol toilet maka penyandang tuna netra dapat mengetahui jika di sekitarnya terdapat ruang toilet.

Dalam tugas akhir ini, dirancang sistem pengenalan simbol toilet untuk tuna netra di dalam ruangan berbasis kamera untuk mempermudah penyandang tuna netra mengenali ruang toilet. Kamera diletakkan pada tas bagian depan pengguna dengan sudut pengambilan gambar tegak lurus kedepan, ini bertujuan untuk mempermudah dalam proses pengambilan data visual. Kemudian data yang didapat diolah dengan beberapa metode pengolahan citra seperti *grayscale* dan *SURF*. Hasil pemrosesan data citra akan dikomunikasikan dalam bentuk suara melalui *earphone* pengguna.

Pengujian dilakukan dengan memasang alat pada pengguna dan pada waktu yang berbeda yakni pada jam 08.00 pagi dan jam 08.00 malam dan dilakukan dengan 5 cara yakni yang pertama pengujian pengenalan simbol dengan membandingkan *keypoint* citra yang ditangkap kamera dengan *keypoint* citra referensinya sendiri, pengujian di tempat umum, pengujian dengan menggabungkan seluruh simbol sebagai citra referensi, pengujian kecepatan pengguna menuju toilet ketika menggunakan sistem, dan pengujian jika terdapat gangguan pada simbol. Sistem ini memiliki error rata-rata sebesar 20% dan akurasi rata-rata sebesar 80%. Error pengenalan simbol disebabkan karena jarak tangkap efektif citra kamera yang digunakan pada sistem ini hanya sebesar 1 meter hingga 3 meter.

Kata kunci: simbol toilet, *grayscale*, *SURF*, kamera, *keypoint*



INDOOR TOILET SYMBOL RECOGNITION SYSTEM USING SURF ALGORITHM FOR THE VISUALLY IMPAIRED BASED ON CAMERA

Name : Tonny Feriandi
1st Advisor : Ronny Mardiyanto, ST., MT., Ph.D.
2nd Advisor : Dr.Ir. Djoko Purwanto, M.Eng.

ABSTRACT

Visual sensing is a process visual data processing used as a reference data base to describe visual objects. The process can be used to help blind people, for example by identifying symbol toilet inside the room. Basically the blind people can not to recognize the environment, so with this device the blind people can find out if toilet room around them.

In this final project, we designed a toilet symbol recognition system for blind people in the room based on cameras to facilitate the blind to recognize environment, especially the toilet room. The camera is placed at the front of the bag user with shooting angle perpendicular to the fore, it aims to simplify the process of making visual data. Then the data obtained is processed with some image processing methods such as grayscale and SURF so that the image captured by the camera can be recognized by the presence or absence of toilet symbol. The results of processing image data to be communicated in the form of sound via earphone.

Experiment were conducted by putting the tools to users and at different times, at 08.00 am and 08:00 pm and conducted with 5 ways the first test by comparing keypoint recognition symbol image captured by the camera with its own reference image keypoint, testing in public places, testing by combining all the symbols as a reference image, test the speed of the user to the toilet when using the system and test when symbol have noise. This system has an average error of 20% and an average accuracy of 80%. Symbol recognition error due to the effective distance of the camera image capture used in this system is only 1 meter to 3 meters.

Keywords: toilet symbol, grayscale, SURF, camera, keypoint



DAFTAR ISI

HALAMAN JUDUL	
PERNYATAAN KEASLIAN TUGAS AKHIR	
LEMBAR PENGESAHAN	
ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Metodologi	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	
2.1 Alat Bantu Pengenalan Lingkungan Untuk Tuna Netra	7
2.1.1 <i>Bionic Eye</i>	7
2.1.2 <i>Navigation and Product Recognition</i>	8
2.2 Metode Pengenalan Simbol	7
2.2.1 Surf (<i>Speeded Up Robust Feature</i>).....	9
2.2.2 PCA (<i>Principal Component Analysis</i>).....	11
2.2.3 LBP (<i>Local Binary Pattern</i>)	12
2.2.4 <i>Chi Square</i>	13
2.3 Citra Digital	13
2.3.1 <i>Grayscale</i>	15
2.3.2 <i>Threshold</i>	16
2.3.3 Citra Biner	18
2.3.4 Citra Warna (RGB).....	18
BAB III PERANCANGAN SISTEM	
3.1 Perancangan Software	24
3.1.1 Pengambilan Citra	24
3.1.2 <i>Grayscale</i>	25
3.1.3 SURF (<i>Speeded-Up Robust Feature</i>).....	26
3.1.3.1 <i>Lokalisasi Keypoint</i>	28

3.1.3.2 Deskriptor <i>Keypoint</i>	29
3.1.3.3 Proses <i>Matching</i>	30
3.1.4 Membandingkan Jumlah <i>Keypoint</i>	31
3.1.5 Perintah Suara.....	32
3.2 Perancangan <i>Hardware</i>	32
3.2.1 Kamera Sebagai Sensor Citra	32
3.2.2 <i>Processing unit</i>	33
3.2.3 <i>Earphone</i>	34
BAB IV PENGUJIAN	
4.1 Pengujian Pengenalan Simbol di Dalam Ruangan	37
4.1.1 Pengujian Pengenalan Simbol di dalam Ruangan Waktu Pagi	37
4.1.1.1. Pengujian Tidak Ada Simbol	38
4.1.1.2 Pengujian Ada Simbol.....	39
4.1.2 Pengujian Pengenalan Simbol di dalam Ruangan Waktu Malam	41
4.1.2.1. Pengujian Tidak Ada Simbol	41
4.1.2.2 Pengujian Ada Simbol.....	42
4.2 Pengujian Pengenalan Simbol Toilet Di Tempat Umum.....	44
4.3 Pengujian Pengenalan Simbol Dengan Menggabung Citra Referensi di dalam Ruangan	45
4.3.1 Pengujian Pengenalan Simbol di dalam Ruangan Waktu Pagi	45
4.3.1.1. Pengujian Tidak Ada Simbol	46
4.3.1.2 Pengujian Ada Simbol.....	46
4.3.2 Pengujian Pengenalan Simbol di dalam Ruangan Waktu Malam	48
4.3.2.1. Pengujian Tidak Ada Simbol	48
4.3.2.2 Pengujian Ada Simbol.....	49
4.4 Pengujian Kecepatan Pengguna Menuju Toilet Ketika Menggunakan Sistem.....	51
4.5 Pengujian Jika Terdapat Gangguan Pada Simbol.....	52
BAB V PENUTUP	53
5.1 Kesimpulan.....	53
5.2 Saran.....	53
DAFTAR PUSTAKA	55
LAMPIRAN	57
BIODATA PENULIS	71

DAFTAR TABEL

Tabel 4.1	Perbandingan hasil pengujian waktu pagi tidak ada simbol	38
Tabel 4.2	Perbandingan hasil pengujian waktu pagi ada simbol.....	40
Tabel 4.3	Perbandingan hasil pengujian waktu malam	41
Tabel 4.4	Perbandingan hasil pengujian waktu malam dengan simbol.....	43
Tabel 4.5	Perbandingan hasil pengujian di tempat umum.....	45
Tabel 4.6	Perbandingan hasil pengujian citra gabungan waktu pagi.....	46
Tabel 4.7	Perbandingan hasil pengujian referensi gabungan waktu pagi	48
Tabel 4.8	Perbandingan hasil pengujian citra gabungan waktu malam	49
Tabel 4.9	Perbandingan hasil pengujian referensi gabungan waktu malam	50
Tabel 4.10	Hasil pengujian kecepatan pengguna menuju toilet ketika menggunakan sistem	51
Tabel 4.11	Perbandingan hasil pengujian gangguan pada simbol.....	52



DAFTAR GAMBAR

Gambar 1.1 Penyandang tuna netra	1
Gambar 1.2 Metodologi pengerjaan tugas akhir	3
Gambar 2.1 Ilustrasi <i>Bionic eye</i>	7
Gambar 2.2 Ilustrasi <i>navigation and product recognition</i>	8
Gambar 2.3 Penempatan orientasi	9
Gambar 2.4 Proses penyusunan deskriptor.....	10
Gambar 2.5 Principal Component Analysis	11
Gambar 2.6 Local Binary Pattern	12
Gambar 2.7 Citra-citra hasil operasi local binary pattern.....	13
Gambar 2.8 Aturan koordinat citra digital.....	14
Gambar 2.9 Posisi koordinat citra digital	15
Gambar 2.10 Citra hasil operasi grayscale	16
Gambar 2.11 Citra grayscale	17
Gambar 2.12 Citra hasil operasi threshold	17
Gambar 2.13 Citra grayscale (kiri), citra threshold (kanan)	17
Gambar 2.14 Gradasi citra biner.....	18
Gambar 2.15 Representasi warna RGB pada citra digital	18
Gambar 3.1 Ilustrasi cara kerja sistem	21
Gambar 3.2 Diagram blok sistem	22
Gambar 3.3 Flowchart <i>image processing</i> pada sistem	23
Gambar 3.4 Tampilan <i>software</i>	24
Gambar 3.5 Citra frame.....	24
Gambar 3.6 <i>Flowchart grayscale</i>	25
Gambar 3.7 Hasil pemrosesan <i>grayscale</i>	26
Gambar 3.8 Flowchart SURF.....	27
Gambar 3.9 Hasil pemrosesan SURF.....	27
Gambar 3.10 Non-maxima suppression	28
Gambar 3.11 Deskriptor keypoint.....	30
Gambar 3.12 Proses membandingkan jumlah keypoint	31
Gambar 3.13 Perancangan hardware	32
Gambar 3.14 Kamera Logitech	33
Gambar 3.15 Ilustrasi pengambilan data citra	33
Gambar 3.16 Processing unit	34
Gambar 3.17 Earphone	35
Gambar 4.1 Simbol : (a)pertama, (b)kedua, (c)ketiga, (d)keempat,	
(e)kelima, (f) pria dan wanita.....	37
Gambar 4.2 Hasil pengujian tidak ada simbol waktu pagi	38

Gambar 4.3 Hasil pengujian ada simbol waktu pagi (a) berhasil dan (b) error.....	39
Gambar 4.4 Grafik perbandingan hasil pengujian waktu pagi	40
Gambar 4.5 Hasil pengujian tidak ada simbol waktu malam.....	41
Gambar 4.6 Hasil pengujian ada simbol waktu malam (a) berhasil dan (b) error.....	42
Gambar 4.7 Grafik perbandingan hasil pengujian waktu malam.....	43
Gambar 4.8 Hasil pengujian di tempat umum.....	44
Gambar 4.9 Grafik perbandingan hasil pengujian	44
Gambar 4.10 Simbol : (a)pertama, (b)kedua, (c)ketiga, (d)keempat, (e)kelima	45
Gambar 4.11 Hasil pengujian tidak ada simbol citra gabungan waktu pagi.....	46
Gambar 4.12 Hasil pengujian ada simbol citra gabungan waktu pagi (a) berhasil dan (b) error	47
Gambar 4.13 Grafik perbandingan hasil pengujian citra gabungan waktu pagi.....	47
Gambar 4.14 Hasil pengujian tidak ada simbol citra gabungan waktu malam	48
Gambar 4.15 Hasil pengujian ada simbol citra gabungan waktu malam (a) berhasil dan (b) error	49
Gambar 4.16 Perbandingan hasil pengujian referensi gabungan waktu malam	50
Gambar 4.17 (a) Proses pengujian sistem dan (b) Peta pengujian.....	51
Gambar 4.18 (a) simbol pertama dan (b) simbol kedua.....	52
Gambar 4.19 Grafik perbandingan hasil pengujian gangguan pada simbol. 52	

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penyandang tuna netra memiliki banyak keterbatasan dalam menjalankan kehidupan sehari-hari. Keterbatasan itu didasari dari ketidakmampuan mereka dalam mengenali lingkungan secara baik. Jika dilihat dari jenisnya, tuna netra dibagi menjadi 2 jenis, yang pertama *low vision* atau kurang dapat melihat yang kedua *totally blind* atau buta total. Penderita *low vision* masih dapat dikoreksi dengan menggunakan kacamata sedangkan *totally blind* sebaliknya, belum ada pengobatan dan alat yang mampu mengembalikan penglihatannya.

Badan kesehatan WHO merilis data setidaknya ada 40-45 juta penderita kebutaan (cacat netra)/gangguan penglihatan. Tahunannya tak kurang dari 7 juta orang mengalami kebutaan atau permenitnya terdapat satu penduduk bumi menjadi buta, ironisnya wilayah dan negara miskinlah yang kebanyakan penduduknya mengalami kebutaan dan gangguan penglihatan. Menurut data Direktorat Jenderal Rehabilitasi Sosial yang dirilis dalam <http://rehsos.kemsos.go.id/>, menunjukkan angka kebutaan di Indonesia mencapai 1,5% (paling tinggi di asia) jika dibandingkan dengan Bangladesh 1%, India 0,7%, dan Thailand 0,3%. Artinya jika ada 12 penduduk di dunia buta setiap jam, empat diantaranya berasal dari Asia Tenggara dan dipastikan 1 orangnya dari Indonesia.



Gambar 1.1 Penyandang tuna netra

Pemerintah Indonesia melalui Kementerian Kesehatan RI hanya dapat melakukan penyuluhan dan pencegahan agar tidak terjadi kebutaan sejak dini. Karena memang saat ini belum ada pengobatan yang dapat mengembalikan indra penglihat apabila mengalami kebutaan total. Sehingga banyak penyandang tuna netra yang mengalami kesulitan dalam melakukan aktifitas sehari-hari, khususnya melakukan aktifitas dasar mereka sebagai manusia.

Keterbatasan itu didasari dari ketidakmampuan mereka dalam mengenali lingkungan secara baik. Misalnya mengenali ruang toilet. Untuk menuju kedalam ruang toilet penyandang tuna netra memerlukan bantuan orang lain, padahal kebutuhan tersebut sangatlah penting.

Untuk itu, dalam tugas akhir ini diimplementasikan rancang bangun sistem pengenalan simbol toilet untuk tuna netra, dimana alat ini akan diletakkan pada tas bagian depan pengguna yang berfungsi untuk mempermudah penyandang tuna netra mengenali ruang toilet. Dengan menggunakan teknik pengolahan citra (*Image Processing*) yang dilakukan oleh *Processing Unit* akan mempermudah pengenalan simbol terjadi, hasil pengenalan simbol akan dikomunikasikan dalam bentuk suara yang sesuai dengan simbol yang telah dikenali. Sehingga penyandang tuna netra dapat secara cepat memenuhi kebutuhan dasarnya dan mengurangi ketergantungan orang lain.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah:

1. Bagaimana mengenali jenis-jenis simbol toilet secara akurat
2. Bagaimana mengkomunikasikan hasil pemrosesan data visual kepada pengguna menggunakan suara dengan baik
3. Bagaimana menempatkan kamera, *processing unit*, dan *earphone* agar tidak membebani pengguna

1.3. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Ruang toilet memiliki simbol toilet.
2. Simbol yang dikenali hanya simbol toilet.
3. Terdiri dari 5 jenis simbol toilet, 1 simbol toilet pria dan wanita
4. Lokasi pelaksanaan memiliki pencahayaan yang cukup.
5. Lokasi pelaksanaan berada di dalam ruangan.
6. Jarak Maksimal simbol 3 Meter
7. Citra referensi simbol diinputkan secara manual

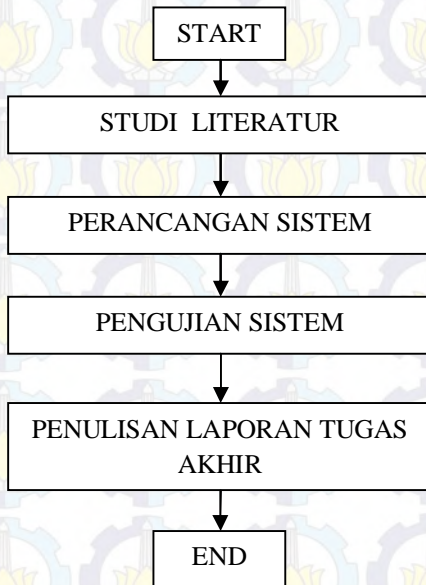
1.4. Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Mampu mengenali jenis-jenis simbol toilet secara akurat.
2. Dapat mengkomunikasikan hasil pemrosesan data citra kepada pengguna menggunakan suara secara baik.
3. Dapat menempatkan kamera, *processing unit*, dan *earphone* secara baik agar tidak membebani pengguna.

1.5. Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut :



Gambar 1.2 Metodologi pengerjaan tugas akhir

STUDI LITERATUR

- Mempelajari prinsip *Image Processing* dalam teknologi *Machine Vision* yaitu akuisisi citra digital dalam pemrosesan citra digital.

- Mempelajari pemrograman visual menggunakan Microsoft Visual C++ 2010 Express dan Open source Computer Vision Library(OpenCV).
- Mempelajari metode yang telah digunakan untuk mengenali simbol-simbol toilet pada jurnal yang telah ada.

PERANCANGAN SISTEM

Setelah mempelajari literatur yang ada, selanjutnya akan dilakukan perancangan sistem. Sistem yang akan dirancang meliputi teknik *image processing* seperti *gray scale* dan *Speeded-Up Robust Feature*.

PENGUJIAN SISTEM

Strategi pengujian sistem dilakukan dengan membandingkan *keypoint* dari citra referensi dengan citra yang ditangkap kamera. Pengujian pengenalan simbol dilakukan di dalam ruangan pada pagi maupun malam hari dengan membandingkan citra simbol referensi dengan citra simbol yang didapat dari kamera, pengujian di tempat umum, pengujian dengan menggabungkan citra referensi menjadi satu, pengujian kecepatan pengguna menuju toilet ketika menggunakan sistem dan pengujian jika terdapat gangguan pada simbol.

PENULISAN LAPORAN TUGAS AKHIR

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang dikerjakan, meliputi pendahuluan, tinjauan pustaka, perancangan sistem, pengujian dan penutup. Apabila hasil yang diperoleh dari tahap pengujian terjadi ketidak sesuaian atau mungkin kurang memuaskan secara bentuk dan ketelitian, maka perlu diadakan evaluasi pada program serta sistem secara keseluruhan. Kemudian dilakukan pengujian ulang sampai parameter keberhasilan telah tercapai.

1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika sebagai berikut:

❖ BAB I : PENDAHULUAN

Pada bagian ini menjelaskan beberapa sub bagian yang antara lain berisi Latar Belakang, Permasalahan, Tujuan, Penelitian pada tugas akhir ini

bertujuan untuk mengenali simbol-simbol toilet untuk membantu tuna netra mengenali ruang toilet. Serta Metodologi, Sistematika Penulisan, dan Relevansi penulisan Tugas Akhir ini.

❖ BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bagian ini berisi mengenai landasan teori yang digunakan dalam pelaksanaan tugas akhir yang meliputi citra digital, *grayscale*, metode *speeded-up robust feature*. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

❖ BAB III : PERANCANGAN SISTEM

Pada bagian ini akan menjelaskan mengenai cara kerja sistem untuk mengenali simbol-simbol toilet.

❖ BAB IV : PENGUJIAN

Pada bagian ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

❖ BAB V : PENUTUP

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangannya.

1.7. Relevansi

Pembuatan alat pengenalan simbol - simbol toilet merupakan salah satu langkah pengembangan dari aplikasi *machine vision* dalam kemajuan teknologi komputer. Pengembangan lebih lanjut dari sistem ini adalah penggunaan pengenalan simbol-simbol selain toilet misalnya simbol ruang makan, keluar, jalan, dan lain-lain, sehingga alat ini mampu mengenali lingkungan yang lebih kompleks.



BAB II

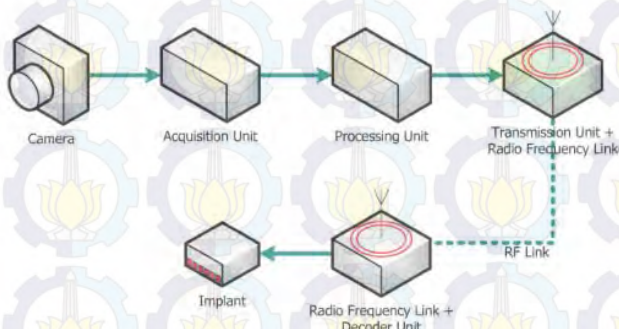
TINJAUAN PUSTAKA DAN DASAR TEORI

Tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya. Sedangkan bagian dasar teori menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem pada tugas akhir ini.

2.1. Alat Bantu Pengenalan Lingkungan Untuk Tuna Netra

2.1.1. *Bionic Eye*

Bionic Eye Merupakan sebuah alat yang menggabungkan proses visual dengan proses biomedik. Pada alat bantu ini terbagi atas 2 proses, yakni proses pengambilan citra dan proses pengiriman citra kepada syaraf otak. Pengambilan citra dilakukan oleh kamera buatan yang dibentuk menyerupai mata, kemudian dikirim kepada *processing unit* untuk di proses, hasil proses dikirim ke syaraf otak menggunakan transmitter RF. berikut adalah ilustrasinya[1]

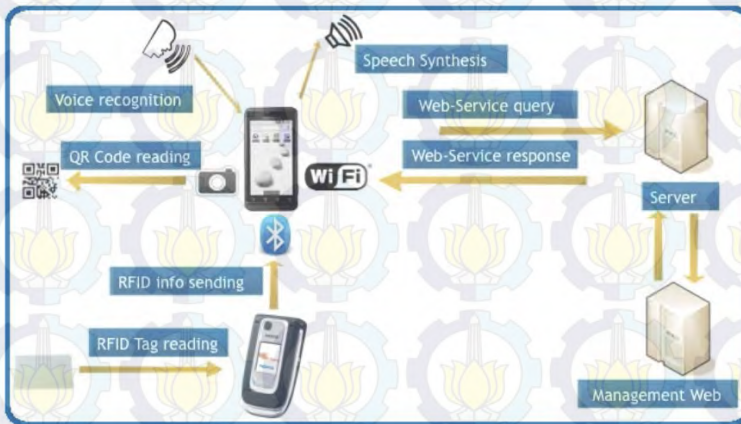


Gambar 2.1 Ilustrasi *Bionic eye*

Pada syaraf mata dipasang elektroda array yang berfungsi untuk mengirimkan data dari pemrosesan citra kepada otak[2], kemudian otak memproses data dari syaraf mata. Sehingga penyandang tuna netra mampu melihat kembali.

2.1.2. Navigation and Product Recognition

Navigation and Product Recognition merupakan salah satu pengenalan lingkungan untuk mempermudah penyandang tuna netra dalam berbelanja dalam hal ini mengenali produk-produk kebutuhan. Alat ini menggunakan bantuan *RFID* dan *Smartphone* sebagai proses pengenalan lingkungannya dan memiliki beberapa proses, proses pertama adalah mencari sebuah produk dengan cara *voice recognition* yang diproses oleh *smartphone*, kemudian *smartphone* memberikan lokasi produk dengan bantuan database perusahaan penyedia layanan tersebut. Setelah didapatkan lokasinya, penyandang tuna netra mencari produk pada lokasi yang sudah ditentukan dengan *RFID*. Setelah didapat, *smartphone* digunakan untuk proses pengenalan produk dengan cara *scanning barcode* yang ada pada produk. berikut adalah ilustrasinya[3].



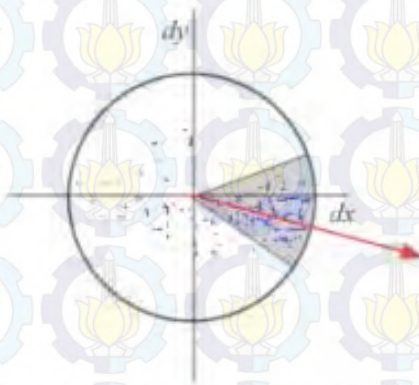
Gambar 2.2 Ilustrasi *navigation and product recognition*

Setelah produk dikenali dengan *barcode scanner* maka, spesifikasi pada produk tersebut dikomunikasikan pada penyandang tuna netra melalui *speaker smartphone*. Sistem alat ini memiliki tahapan dan kerumitan yang cukup kompleks namun dapat digunakan untuk membantu penyandang tuna netra dalam berbelanja dan mengenali produk yang akan dibeli

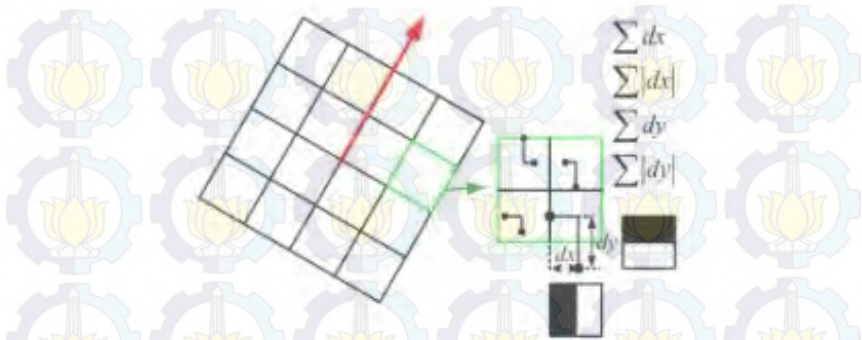
2.2. Metode Pengenalan Simbol

2.2.1. Surf (*Speeded Up Robust Feature*)[4]

SURF ini merupakan *keypoint* dari sebuah gambar. *Keypoint* adalah titik-titik dari sebuah gambar yang nilainya tetap ketika mengalami perubahan skala, rotasi, *blurring*, pencahayaan, dan juga perubahan bentuk. Berikut merupakan proses untuk mendapatkan metode *surf*, yang pertama adalah *Integral Image*. Dasar dari algoritma SURF adalah *integral image*. *integral image* digunakan untuk menghitung *box filter* dan *Haar filter response* di skala yang berubah-ubah pada waktu yang konstan tiap pikselnya[4]. Yang kedua adalah *Interest Point Detection*, *Interest Point Detection* merupakan proses untuk mendapatkan titik-titik *keypoint* yang selanjutnya digunakan sebagai pembanding. Proses yang ketiga adalah *Interest Point Descriptor*, Untuk mendapatkan *interest point descriptor* terdapat dua langkah, yang pertama adalah menghitung orientasi utama di sekitar *interest point*. Pertama-tama menyusun wilayah berbentuk lingkaran sebesar $6s$ (s adalah skala yang didapat dari *interest point*) jarak dari sekitar *interest point* yang diketahui ditujukan untuk menghasilkan orientasi reproduisibel (seperti gambar 2.3).



Gambar 2.3 Penempatan orientasi



Gambar 2.4 Proses penyusunan deskriptor

Step yang kedua (seperti gambar 2.4), *interest point descriptor* disusun dari mengekstrak kotak *window* yang berada disekitar *interest point*. Windownya di bagi menjadi 4x4 kotak tiap sub-wilayah dengan wilayah *sample point* didalam yang umumnya berukuran 5x5. Di tiap sub-wilayah, dihitung respon dari *Haar wavelet* pada arah horisontal dan vertikal (dx dan dy) dan responsnya berasal dari bobot tengah *Gaussian* pada *interest point*. Dimana setelahnya, *wavelet response* sepanjang x dan y dihitung pada tiap sub wilayah. Proses yang keempat adalah *Feature Matching*, Mencocokkan dua fitur dapat dilakukan dengan membandingkan *corresponding interest point descriptor*, proses pencocokan melalui tanda dari *Laplacian* yang menjadi dasar dari *interest point*. *Laplacian correspond* mengikuti dari matriks *Hessian*. Karena itu, maka ketika mulai tahap pengenalan pada proses ini telah terjadi perhitungan algoritma dan hanya perlu dibandingkan *interest point descriptor* nya ketika telah memiliki tanda yang sama dan kontras yang sama. Lalu, menghitung jarak *Euclidian* antara dua *interest point descriptor* dan akan cocok jika jarak *Euclidean* pada wilayah deskriptor mendekati 0.8 kali jarak dari *nearest neighbor* kedua. (0.8 adalah *threshold*, bisa diubah, semakin kecil nilainya semakin kecil pula titik kecocokan yang didapat) [5].

2.2.2. PCA (Principal Component Analysis)[6]

Prosedur *PCA* pada dasarnya adalah bertujuan untuk menyederhanakan variabel yang diamati dengan cara menyusutkan (mereduksi) dimensinya. Hal ini dilakukan dengan cara menghilangkan korelasi di antara variabel bebas melalui transformasi variabel besar asal ke variabel baru yang tidak berkorelasi sama sekali atau yang biasa disebut dengan *principal component*.

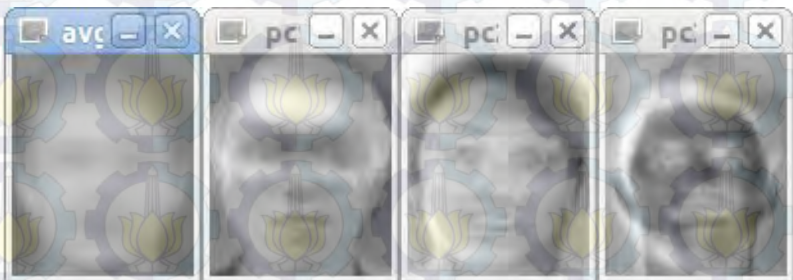
Umumnya metode ini digunakan untuk mengenali wajah, dimana data wajah diambil beberapa kali dari tiap beberapa posisi, kemudian data yang berukuran besar tersebut di ubah menjadi data yang lain dengan ukuran lebih kecil.

Eigen face adalah salah satu algoritma pengenalan wajah yang didasarkan pada *PCA*. Untuk menghasilkan *eigenface*, sekumpulan besar citra digital dari wajah manusia di normalisasi dan kemudian diolah pada resolusi yang sama dan kemudian diperlakukan sebagai vektor dimensi.

Sebelum mengaplikasikan metode *PCA* dalam hal ini mereduksi dimensi dari vektor fitur yang ada maka nilai *eigen* dari citra simbol yang ada perlu diurutkan terlebih dahulu sehingga nantinya didapat nilai *eigen* yang tereduksi yang diinginkan. Setiap *Principal component* merupakan representasi dari suatu kombinasi linier dari semua citra-citra simbol training yang sudah dikurangi oleh citra rata-rata. Gabungan citra simbol-simbol inilah yang dinamakan dengan *eigenvalue*. *Eigenvalue* - *eigenvalue* ini merupakan fitur dari suatu simbol yang nantinya akan dikenali. Jika λ merupakan nilai *eigen* dan berkoresponden dengan vektor ortogonal X , dimisalkan contohnya adalah A maka,

$$Y_k = AX_k \quad (k=1,2,...,d)$$

Dimana Y adalah komponen dari sampel gambar.



Gambar 2.5 *Principal component analysis*

2.2.3. LBP (Local Binary Pattern)[7]

Local Binary Pattern (LBP) adalah deskriptor tekstur yang umumnya digunakan untuk mengenali wajah, namun prinsipnya bisa digunakan untuk mengenali simbol. Karena selain wajah, simbol juga memiliki komposisi *micro-texture-pattern* yaitu suatu operator non parametrik yang menggambarkan tat ruang lokal suatu citra. *LBP* didefinisikan sebagai perbandingan nilai biner piksel pada pusat citra dengan 8 nilai piksel di sekelilingnya. Misal pada sebuah citra berukuran 3x3, nilai biner pada pusat citra dibandingkan dengan nilai sekelilingnya. Nilai sekelilingnya akan bernilai 1, jika nilai piksel pusat lebih kecil dan bernilai 0 jika nilai biner pusat lebih besar. setelah itu, menyusun 8 nilai biner 8 searah jarum jam atau sebaliknya dan merubah 8 bit biner kedalam nilai decimal untuk menggantikan nilai piksel pada pusat.



Gambar 2.6 Local binary pattern

Nilai desimal dari 8 bit dinyatakan dalam persamaan berikut :

$$LBP_{P,R}(X_C, Y_C) = \sum_{p=0}^{P-1} S(gp - gc) 2^p$$

Dan fungsi $s(x)$ didefinisikan sebagai berikut :

$$s(X) = \begin{cases} 1, & X \geq 0 \\ 0, & X < 0 \end{cases}$$



Gambar 2.7 Citra-citra hasil operasi *local binary pattern*

2.2.4. *Chi Square*[8]

Chi square digunakan sebagai *classifier* dalam ruang fitur. Pengujian pengenalan citra dilakukan dengan cara menguji kemiripan dari 2 buah distribusi, yaitu distribusi dari histogram citra test dan distribusi dari histogram citra-citra training. Prinsip dasar *Chi Square* adalah dengan menghitung nilai minimum dari rata-rata histogram citra untuk menentukan kemiripan dari 2 citra.

$$X^2(S,M) = \sum_i \frac{(S_i - M_i)^2}{|S_i + M_i|}$$

Untuk daerah spatial dihitung dengan cara menjumlah *i* (bin dari histogram) dan *j* (region citra).

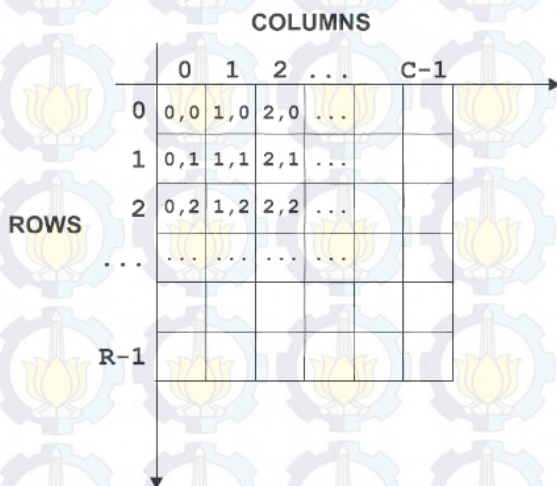
$$X^2_w(S,W) = \sum_i \sum_j W_j \frac{(S_i - M_i)^2}{|S_i + M_i|}$$

Dimana W_j adalah bobot untuk setiap region *j*.

2.3. Citra Digital

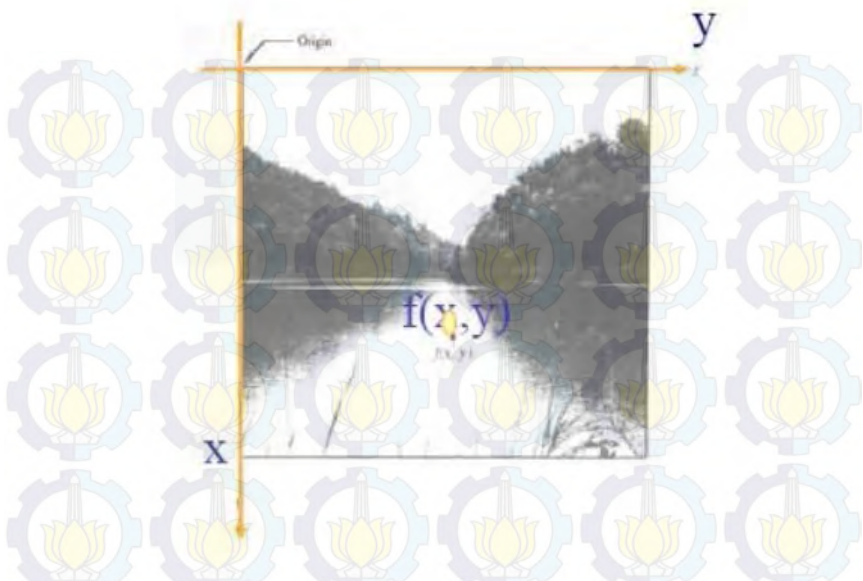
Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang

dapat langsung disimpan pada suatu media penyimpan. Menurut arti secara harfiah, citra adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya. Pantulan cahaya ini ditangkap oleh alat-alat optik, seperti mata pada manusia, kamera, pemindai, dan lain-lain sehingga bayangan objek dalam bentuk citra dapat terekam. Citra yang diolah menggunakan processing unit merupakan citra digital, biasanya citra tersebut berbentuk file.



Gambar 2.8 Aturan koordinat citra digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ yang merupakan intensitas citra pada koordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*). Sistem koordinat pada sebuah citra digital dapat dilihat pada Gambar 2.9.[9]



Gambar 2.9 Posisi koordinat citra digital

2.3.1. *Grayscale*

Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B. [10]



Gambar 2.10 Citra hasil operasi *grayscale*

2.3.2. *Threshold*

Citra *threshold* dilakukan dengan mempertegas citra dengan cara mengubah citra hasil yang memiliki derajat keabuan 255 (8 bit), menjadi hanya dua buah yaitu hitam dan putih. Hal yang perlu diperhatikan pada proses *threshold* adalah memilih sebuah nilai *threshold* (T) dimana piksel yang bernilai dibawah nilai *threshold* akan diset menjadi hitam dan piksel yang bernilai diatas nilai *threshold* akan diset menjadi putih.

Jika $f(x,y)$ adalah nilai intensitas pixel pada posisi (x,y) maka pixel tersebut diganti putih atau hitam tergantung kondisi berikut :

$$f(x,y) = 255, \text{ jika } f(x,y) \geq T$$

$$f(x,y) = 0, \text{ jika } f(x,y) < T$$

Sebagai contoh misalnya diketahui citra *grayscale* 4x4 pixel dengan kedalaman 8 bit seperti Gambar 2.11.

200	230	150	75
240	50	170	92
210	100	120	80
100	90	200	230

Gambar 2.11 Citra *grayscale*

Bila nilai $T = 145$ diterapkan untuk citra pada Gambar 2.11 di atas maka diperoleh citra seperti pada Gambar 2.12.[11]

255	255	255	0
255	0	255	0
255	0	0	0
0	0	255	255

Gambar 2.12 Citra hasil operasi *threshold*



Gambar 2.13 Citra *grayscale* (kiri), citra *threshold* (kanan)

2.3.3. Citra Biner

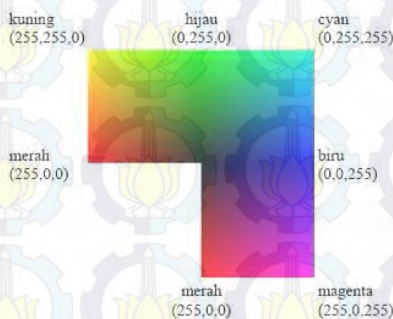
Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (background) dan 1 menyatakan warna tinta/objek atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra *threshold* sebelumnya. Gradasi citra biner dapat dilihat pada Gambar 2.14. [12]



Gambar 2.14 Gradasi citra biner

2.3.4. Citra Warna (*RGB*)

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 1677726$ jenis warna.



Gambar 2.15 Representasi warna *RGB* pada citra digital

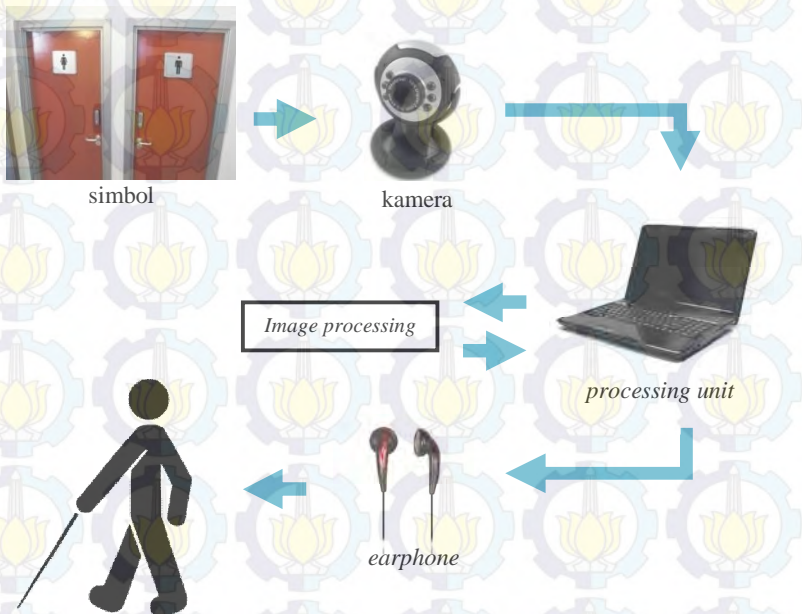
Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen *Red*, *Green*, *Blue*. Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = $RGB(30, 75, 255)$. Putih = $RGB(255,255,255)$, sedangkan untuk hitam= $RGB(0,0,0)$. [13]



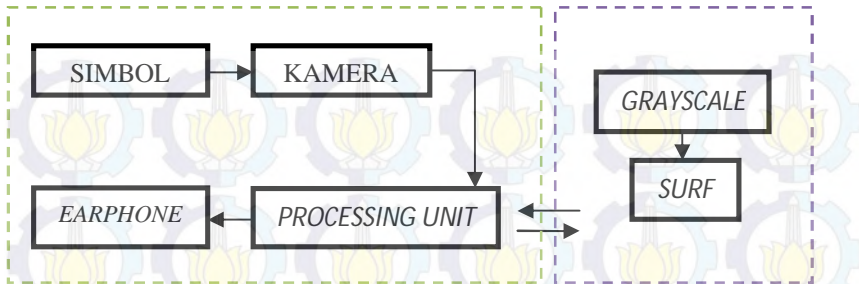
BAB III

PERANCANGAN SISTEM

Pada tugas akhir ini, sistem dilakukan di sebuah *processing unit* berupa *notebook*. Input data citra didapat dari sensor kamera berupa *webcam*. Sensor kamera diletakkan pada tas bagian depan pengguna. Hal ini dilakukan untuk mempermudah pengambilan data citra yang ada didepan pengguna. Kemudian dilakukan pengolahan citra gambar untuk mengenali ada tidaknya simbol di depan pengguna. Ketika simbol ditemukan maka akan dikomunikasikan dalam bentuk suara. Ilustrasi cara kerja sistem ditunjukkan pada gambar 3.1



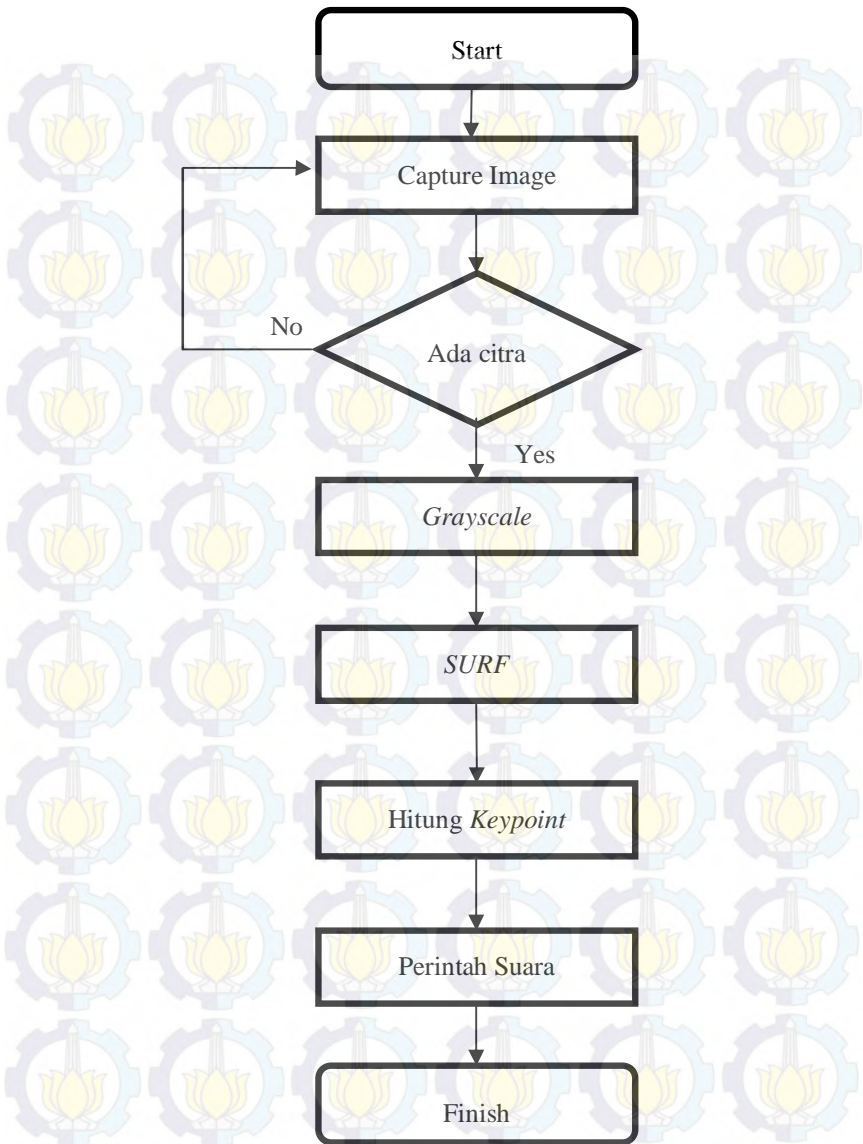
Gambar 3.1 Ilustrasi cara kerja sistem



Gambar 3.2 Diagram blok sistem

Berikut ini merupakan cara kerja keseluruhan sistem :

- a. Memastikan *processing unit* telah aktif memproses.
- b. Kamera menangkap citra yang berada di depan pengguna, kemudian data citra diolah oleh *processing unit* dengan metode-metode pengolahan citra.
- c. Metode pengolahan citra antara lain dilakukan proses *grayscale* dan *SURF* (*Speeded-Up Robust Feature*) untuk mengenali ada tidaknya simbol. Apabila simbol dikenali maka akan muncul banyak *keypoint* pada citra input.
- d. Memberikan nilai pada data *SURF* untuk memberikan batas kesamaan antara data dari input citra dengan data dari citra simbol.
- e. Pengenalan simbol dilakukan dengan cara membandingkan jumlah *keypoint* pada input citra dengan data dari citra simbol.
- f. Memberikan batasan jumlah *keypoint* yang terdapat pada input citra untuk referensi kapan perintah suara bekerja.
- g. Apabila jumlah *keypoint* memenuhi keberadaan simbol, maka dikomunikasikan melalui perintah suara.

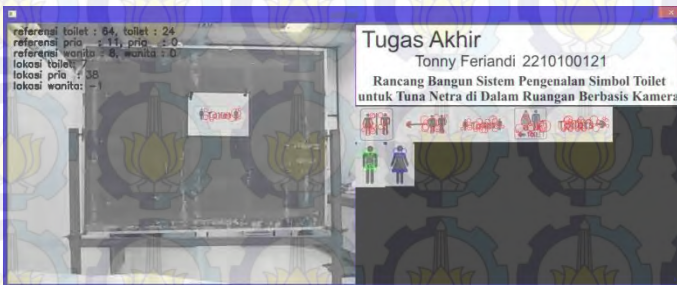


Gambar 3.3 Flow chart *image processing* pada sistem

Berikut ini adalah penjelasan dari keseluruhan sistem.

3.1. Perancangan Software

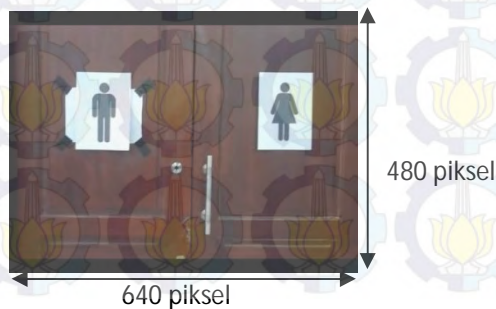
Tahapan ini terdiri dari beberapa tahapan yakni, pengambilan citra, *grayscale*, *surf*, membandingkan jumlah *keypoint*, perintah suara. Setiap tahapan memiliki fungsi masing-masing yang saling mendukung serta memiliki tahapan yang berurutan.



Gambar 3.4 Tampilan software

3.1.1. Pengambilan Citra

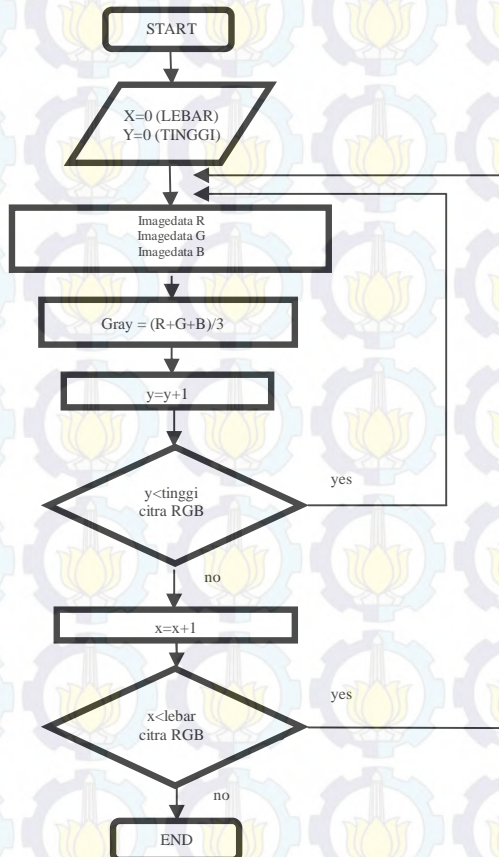
Data citra simbol yang didapat pada input citra disimpan dalam bentuk variabel frame. Variabel inilah yang akan diolah dengan metode pengenalan simbol untuk mengenali ada tidaknya simbol yang didapat dari input citra. Pada sistem ini data citra yang disimpan memiliki ukuran 640x480 piksel.



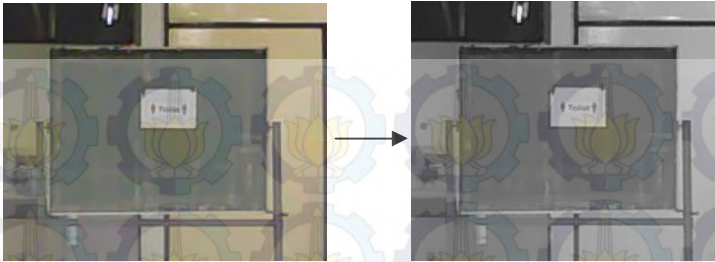
Gambar 3.5 Citra frame

3.1.2. Grayscale

Langkah pertama setelah data citra didapat adalah mengubah gambar *RGB*, kedalam bentuk *grayscale*. Proses ini dilakukan karena *RGB* memiliki 24 bit, dimana masing-masing kanal warna memiliki nilai 8 bit. Jika tidak dikonversi kedalam bentuk *grayscale*, maka perhitungan nilai piksel akan sangat sulit, karena metode *surf* memerlukan frame yang bernilai 8 bit untuk mengurangi perhitungan yang kompleks.



Gambar 3.6 Flowchart grayscale



Gambar 3.7 Hasil pemrosesan *grayscale*

Pada tugas akhir ini, digunakan fungsi yang sudah ada dalam library *opencv* untuk mengubah dari RGB ke *grayscale*. berikut adalah prosedurnya :

```
cvCvtColor( object, object_color, CV_RGB2GRAY );
```

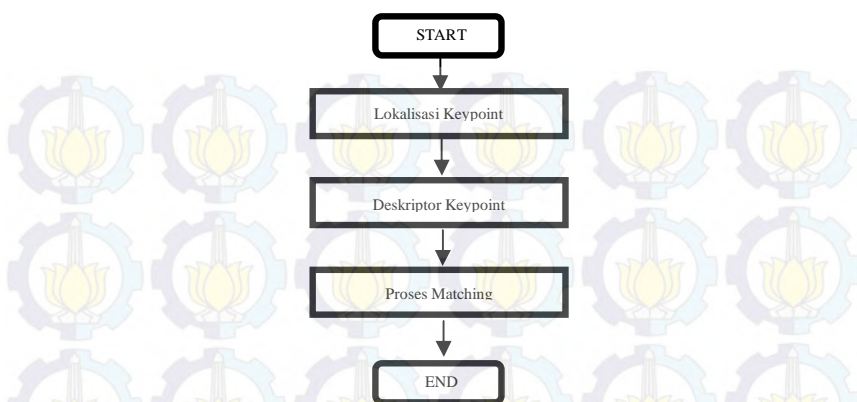
object : merupakan sumber frame yang akan di konversi
object_color : merupakan frame yang akan dipakai sebagai wadah hasil konversi
CV_RGB2GRAY : Perintah untuk mengkonversi dari RGB ke GRAY

Secara Manual Mengubah RGB kedalam bentuk Grayscale dalam dilakukan dengan cara menjumlah data gambar tiap kanal R,G, dan B dan dibagi 3.

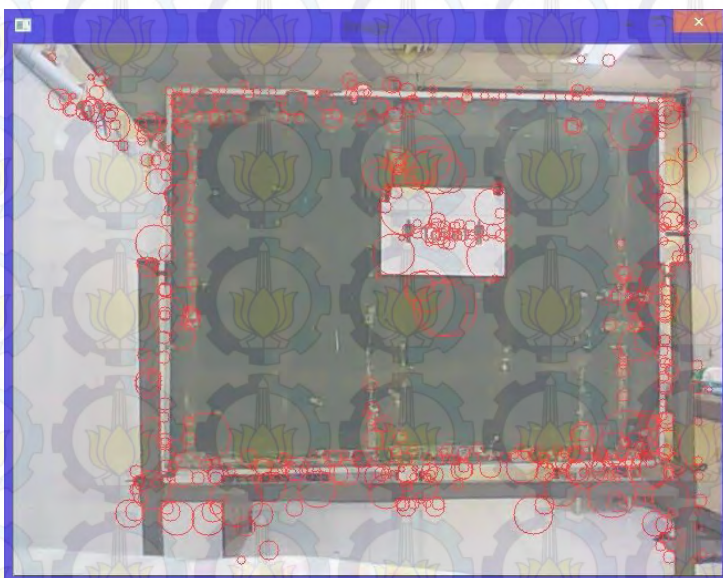
$$I_{grayscale} = \frac{Ir+Ig+Ib}{3}$$

3.1.3. SURF (Speeded-Up Robust Feature)

Metode *surf* merupakan metode yang menggunakan *keypoint* dari sebuah gambar sebagai referensinya. *Keypoint* adalah titik-titik dari gambar yang nilainya akan sama jika mengalami perubahan skala, rotasi, *blurring*, pencahayaan dan juga bentuk. Pada metode *surf* memiliki beberapa tahapan yakni, lokalisasi *keypoint*, deskriptor *keypoint* dan proses matching.



Gambar 3.8 *Flowchart SURF*

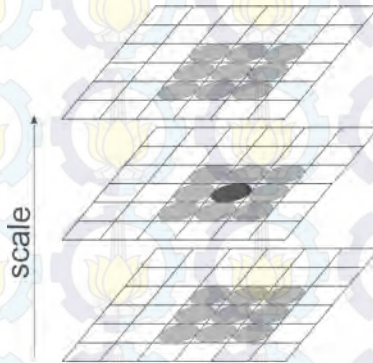


Gambar 3.9 Hasil pemrosesan *SURF*

3.1.3.1. Lokalisasi Keypoint

Keypoint adalah titik-titik yang dihasilkan dari metode *surf*. Pada metode ini *keypoint* digunakan sebagai karakteristik suatu citra atau gambar. Sehingga setiap gambar atau citra memiliki jumlah *keypoint* yang berbeda. Lokalisasi *keypoint* dilakukan dengan beberapa proses. Proses pertama menentukan *threshold* untuk *keypoint*.

Proses berikutnya *non-maxima suppression*, proses ini merupakan proses mencari calon *keypoint* dengan membandingkan tiap-tiap piksel gambar pada *scale space* dengan 26 tetangga yang terdiri dari 8 titik di *scale* asli dan 9 titik di *scale* atas bawahnya. Proses inilah yang menghasilkan *keypoint*, gambar 3.10 menunjukkan *non-maxima suppression*



Gambar 3.10 *Non-maxima suppression*

Proses terakhir adalah mencari lokasi *keypoint* menggunakan interpolasi data yang dekat dengan *keypoint* hasil proses sebelumnya. Untuk mencari nilainya dapat dilakukan dengan determinan *Hessian* $H(x,y,\sigma)$ seperti berikut ini[14]:

$$H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 H}{\partial x^2} x$$

Lokasi ekstrim yang diinterpolasi ditemukan dari turunan fungsi diatas, sehingga menjadi :

$$x = -\frac{\partial^2 H}{\partial x^2} \frac{\partial H}{\partial x}$$

Pada tugas akhir ini, digunakan fungsi yang sudah ada dalam library opencv untuk mendapatkan lokasi *keypoint*. berikut adalah prosedurnya :

```
CvSURFPoint*r=(CvSURFPoint*)cvGetSeqElem(imageKeypoints,i);
```

imageKeypoints : *Keypoint* dari hasil ekstrak yang akan dicari lokasinya
r : Wadah yang digunakan untuk nilai dari lokasi koordinat *keypoint*
i : Menunjukkan *keypoint* beberapa yang dicari lokasinya

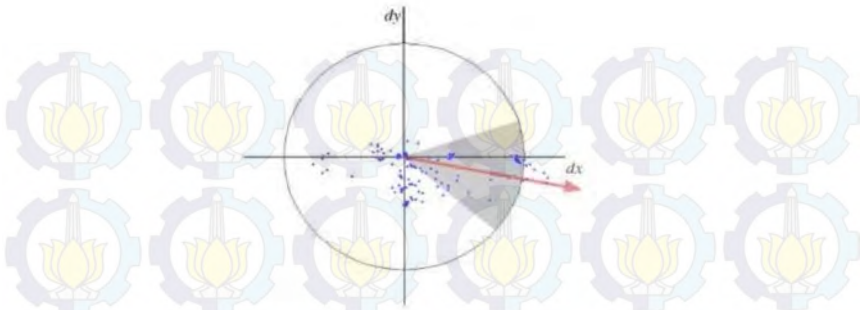
3.1.3.2. *Deskriptor Keypoint*

Deskriptor merupakan daerah piksel disekitar *keypoint* yang dihasilkan. Deskriptor menggambarkan sebaran intensitas piksel tetangga di sekitar *keypoint*. Sehingga deskriptor sendiri dapat dikatakan sebagai ciri dalam sebuah citra karena tiap citra memiliki sebaran intensitas piksel yang berbeda.

Deskriptor *keypoint* terdiri dari dua proses perhitungan, yang pertama adalah Pembuatan Orientasi. Pembuatan orientasi dilakukan dengan menghitung respon *Haar Wavelet* arah x dan y pada daerah piksel tetangga *keypoint* yang berjarak 6 *scale*. Orientasi dominan dihitung menggunakan jumlah semua respon pada pergeseran orientasi dengan ukuran $\pi/3$ seperti gambar 3.11, kemudian respon horisontal dan vertikal dijumlahkan sehingga menghasilkan sebuah vektor orientasi lokal. Vektor terpanjang inilah merupakan orientasi *keypoint*.

Proses kedua adalah berdasarkan jumlah respon *Haar Wavelet*, pada proses ini dilakukan dengan membuat daerah kotak disekitar *keypoint*, dimana titik pusatnya *keypoint* itu sendiri. Kemudian, respon *wavelet* dx dan dy dijumlahkan pada setiap *sub-region*. Maka akan didapatkan nilai absolut dari respon *wavelet*. Masing-masing *sub-region* memiliki 4 dimensi deskriptor sehingga vektornya berjumlah 64. Maka didapat[14] :

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right)$$



Gambar 3.11 Deskriptor *keypoint*

Pada tugas akhir ini, deskriptor dapat dicari menggunakan fungsi yang sudah ada dalam library opencv. berikut adalah prosedurnya :

```
cvExtractSURF( image, 0, &imageKeypoints, &imageDescriptors,
               storage, params );
```

image : frame yang akan di ekstrak
&imageKeypoints : Perintah untuk mendapatkan keypoint
&imageDescriptors : Perintah untuk mendapatkan deskriptor
storage : wadah untuk menampung nilai keypoint dan deskriptor sementara
params : Parameter yang digunakan untuk menangkap nilai keypoint dan deskriptor pada sistem ini nilai params = 500

3.1.3.3. *Proses Matching*

Pada sistem ini proses matching dilakukan menggunakan teknik *FLANN (Fast Library Approximated Nearest Neighbor)*. Fitur *SURF* ini terdiri dari *keypoint* dan deskriptor yang berupa vektor. Pada intinya sistem *matching FLANN* adalah dengan mencari data dari vektor keypoint dan deskriptor pada citra yang didapat dari kamera lalu membandingkannya dengan data vektor keypoint dan deskriptor pada citra referensi, sehingga nilai vektor pada citra yang didapat dari kamera yang mendekati nilai vektor pada citra referensi akan dianggap cocok atau sama, Pada sistem ini jika nilai vektor dari citra kamera kurang dari 0,6 dari citra referensi maka dianggap sama.

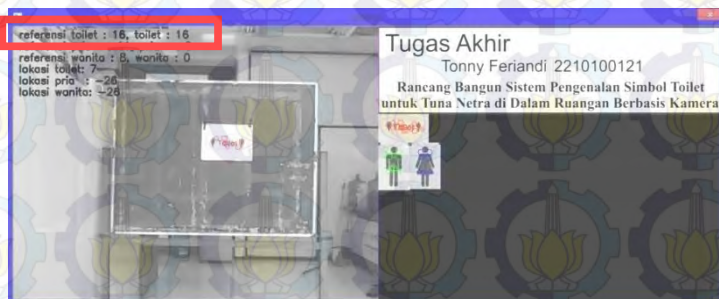
```

int naiveNearestNeighbor(const float* vec, int laplaci an, const
CvSeq* model_keypoints, const CvSeq* model_descriptors )
{
    int length =(int)(model_descriptors->elem_size/sizeof (float));
    int i, neighbor = -1; double d, dist1 = 1e6, dist2 = 1e6;
    CvSeqReader reader, kreader;
    cvStartReadSeq( model_keypoints, &kreader, 0 );
    cvStartReadSeq( model_descriptors, &reader, 0 );
    for( i = 0; i < model_descriptors->total; i++ )
    {
        const CvSURFPoint* kp = (const CvSURFPoint*)kreader.ptr;
        const float* mvec = (const float*)reader.ptr;
        CV_NEXT_SEQ_ELEM( kreader.seq->elem_size, kreader );
        CV_NEXT_SEQ_ELEM( reader.seq->elem_size, reader );
        if( laplaci an != kp->laplaci an )continue;
        d = compareSURFDescriptors( vec, mvec, dist2, length );
        if( d < dist1 ){dist2 = dist1; dist1 = d; neighbor = i;}
        else if ( d < dist2 )dist2 = d;
    }
    dist=dist+dist1;
    if ( dist1 < 0.6*dist2 )return neighbor; return -1; }

```

3.1.4. Membandingkan Jumlah Keypoint

Proses ini dilakukan untuk mendapatkan referensi ada tidaknya simbol. Dari proses matching sebelumnya didapat *keypoint* pada kedua gambar, *keypoint* ini digunakan sebagai acuan dari proses membandingkan jumlah *keypoint*. Apabila jumlah *keypoint* pada gambar pertama mendekati jumlah *keypoint* pada gambar kedua maka akan disimpulkan ada simbol begitu juga sebaliknya jika jumlah *keypoint* pertama dan kedua terlampaui jauh tidak sama maka akan disimpulkan tidak ada simbol.



Gambar 3.12 Proses membandingkan jumlah *keypoint*

3.1.5. Perintah Suara

Perintah suara pada sistem ini digunakan untuk mempermudah mengkomunikasikan hasil proses pengenalan simbol kepada pengguna. perintah suara pada tugas akhir ini menggunakan fungsi yang disediakan oleh visual studio, berikut adalah prosedurnya :

```
PlaySound(TEXT("toi let.wav"), NULL, SND_SYNC);
```

3.2. Perancangan Hardware

Perancangan hardware pada sistem ini terdiri dari *web* kamera, *processing unit* dan *earphone*.



Gambar 3.13 Perancangan hardware

3.2.1. Kamera Sebagai Sensor Citra

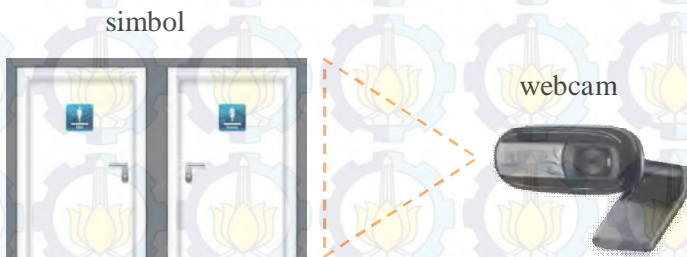
Kamera yang digunakan sebagai sensor citra adalah *Logitech Webcam C170*. Dimana spesifikasinya adalah sebagai berikut :

- High-Resolution Snapshots Up to 5 megapixel
- Perekaman video hingga 1024 x 768 pixel
- Hi-Speed USB 2.0
- Auto Colour Compensation dan Auto Brightness Adjustment
- Auto Focus
- Lens Sandard (Premium Lens)



Gambar 3.14 Kamera Logitech C170[15]

Hal terpenting dalam pemilihan sensor citra pada sistem ini adalah kamera memiliki image sensor yang tinggi agar diperoleh gambar yang tajam, serta memiliki frame rate yang tinggi. Hal tersebut dilakukan untuk membantu proses pengolahan citra agar mengurangi beban *processing unit* serta kualitas citra yang baik akan menghasilkan akurasi yang baik pula. Kamera diletakkan di depan tubuh pengguna untuk mempermudah pengenalan simbol.



Gambar 3.15 Ilustrasi pengambilan data citra

3.2.2. Processing unit

Pada sistem ini dipilih *notebook* sebagai *processing unit* karena beberapa alasan yaitu, *Notebook* bersifat *portable*, spesifikasi tinggi, memiliki ukuran yang cukup kecil, tidak terlalu berat. *Notebook* juga memiliki baterai bawaan yang cukup mampu digunakan berjam-jam dan memiliki banyak port usb yang penting untuk sistem ini.



Gambar 3.16 *Processing unit*[16]

Dimana spesifikasinya adalah sebagai berikut :

- a. Prosesor intel core i3 1.8ghz Hyperthread (4 core)
- b. Ram 8 Gb
- c. Hardisk Internal 500 Gb
- d. port Speaker
- e. Batrei 4Cells 2950 mAh 44 Whrs
- f. Berat 2.0 kg
- g. Ultra slim (ketebalan 21mm)
- h. Windows 8

Hal terpenting dalam pemilihan *processing unit* dalam sistem ini adalah kecepatan *clock processor* dan memory ram yang mencukupi. Ini dikarenakan pengolahan citra membutuhkan memory ram yang besar dan pemrosesan yang cepat. Apabila di bawah standar, pemrosesan akan mengalami hambatan seperti *lagging* dan *break* saat proses berlangsung .

3.2.3. Earphone

Earphone digunakan untuk mempermudah mengkomunikasikan hasil pengenalan simbol kepada pengguna, pada sistem ini digunakan sennheiser mx 170 sebagai output suaranya.



Gambar 3.17 *Earphone*[17]

Dimana spesifikasinya adalah sebagai berikut :

- a. Frekuensi 22-20.000 Hz
- b. Impedansi 32 ohm
- c. Kepekaan 109 dB
- d. Berat 14g



BAB IV PENGUJIAN

Pada tugas akhir ini pengujian dilakukan untuk mengetahui kinerja sistem. Diantaranya, pengujian pengenalan simbol di dalam ruangan pada pagi maupun malam hari dengan membandingkan citra simbol referensi dengan citra simbol yang didapat dari kamera, pengujian di tempat umum, pengujian dengan menggabungkan citra referensi menjadi satu, pengujian kecepatan pengguna menuju toilet ketika menggunakan sistem dan pengujian jika terdapat gangguan pada simbol. Pengujian pengenalan simbol toilet dengan metode *surf*. Bentuk pengujiannya adalah membandingkan *keypoint* dari citra referensi dengan citra input kamera pada jarak 1 meter hingga 5 meter sehingga didapat nilai *error*nya. Pengujian dilakukan di dalam ruangan laboratorium B202 teknik elektro ITS dan di tempat umum di Surabaya. Pengujian di dalam laboratorium B202 teknik elektro ITS dilakukan saat pagi dan malam hari dengan pencahayaan yang cukup, sedangkan pengujian di tempat umum dilakukan pada pukul 12.00 siang dengan pencahayaan yang cukup.

4.1. Pengujian Pengenalan Simbol di Dalam Ruangan

Pengujian pengenalan simbol yang pertama dilakukan dengan cara membandingkan *keypoint* tiap simbol dengan citra referensinya adalah simbol itu sendiri. Hal ini dilakukan untuk mengetahui jumlah *keypoint* pada citra referensi dan pada citra yang didapat dari kamera.

4.1.1. Pengujian Pengenalan Simbol di dalam Ruangan Waktu Pagi

Pengujian dilakukan pada jam 08.00 pagi. Ruangan memiliki sirkulasi cahaya yang baik sehingga pada jam tersebut sinar matahari sudah menerangi ruangan. Kamera diletakkan dengan posisi tegak lurus menghadap simbol.

4.1.1.1. Pengujian Tidak Ada Simbol

Pengujian tidak ada simbol dilakukan untuk mengetahui ada tidaknya *keypoint* ketika didepan kamera tidak ada simbol, jika ada *keypoint* maka terdapat *error*.

4.1.1.2. Pengujian Ada Simbol

Dari percobaan yang dilakukan didapatkan hasil dengan *error* rata-rata untuk perhitungan dari jarak 1 meter hingga 5 meter adalah 25.20%. *Error* terjadi dikarenakan jarak efektif kamera menangkap citra simbol hanya 1-3 meter. Hal ini menyebabkan terjadi perubahan bentuk piksel pada citra yang berada di atas 3 meter sehingga menyebabkan sebagian citra simbol tidak dapat dikenali. Namun *error* ini dapat diatasi dengan cara mengganti kamera *web* yang memiliki resolusi dan *image* sensor yang lebih tinggi. Pengujian dilakukan terhadap 5 jenis simbol toilet dan masing-masing 1 simbol toilet pria dan wanita

Dari data diatas dapat diketahui *error* pada simbol pertama 11.11%, kedua 29.17%, ketiga 29.17%, Keempat 19.6%, Kelima 13.13%, Pria 35.35% , Wanita 38.88%. Sehingga didapat *error* rata-rata sebesar 25.20%.

4.1.2. Pengujian Pengenalan Simbol di dalam Ruang Waktu Malam

Pengujian dilakukan pada jam 20.00. Pada jam tersebut matahari sudah terbenam sehingga ruangan diberi penerangan melalui cahaya lampu. Kamera diletakkan dengan posisi tegak lurus menghadap simbol.

4.1.2.1. Pengujian Tidak Ada Simbol

Pengujian tidak ada simbol dilakukan untuk mengetahui ada tidaknya *keypoint* ketika didepan kamera tidak ada simbol, jika ada maka terdapat *error*.

Dari data diatas terlihat bahwa pengujian tanpa simbol memiliki *error* lebih besar dari pada pagi hari yakni sebesar 7.22 %.

4.1.2.2. Pengujian Ada Simbol

Dari percobaan yang dilakukan didapatkan hasil dengan *error* rata-rata untuk perhitungan dari jarak 1 meter hingga 5 meter adalah 25.20%. *Error* terjadi dikarenakan jarak efektif kamera menangkap citra simbol hanya 1-3 meter. Hal ini menyebabkan terjadi perubahan bentuk piksel pada citra yang berada di atas 3 meter sehingga menyebabkan sebagian citra simbol tidak dapat dikenali. Namun *error* ini dapat diatasi dengan cara mengganti kamera web yang memiliki resolusi dan *image* sensor yang lebih tinggi. Pengujian dilakukan terhadap 5 jenis simbol toilet dan masing-masing 1 simbol toilet pria dan wanita.

Dari data diatas dapat diketahui *error* pada simbol pertama 11.11%, kedua 29.17%, ketiga 29.17%, Keempat 19.6%, Kelima 13.13%, Pria 35.35% , Wanita 38.88%. Sehingga didapat *error* rata-rata sebesar 25.20%.

4.2. Pengujian Pengenalan Simbol Toilet Di Tempat Umum

Pengujian di tempat umum dilakukan menggunakan video hasil perekaman di tempat umum. Dari percobaan yang dilakukan didapatkan hasil dengan *error* rata-rata untuk perhitungan dari jarak 1 meter hingga 5 meter adalah 22.22%. *Error* terjadi dikarenakan jarak efektif kamera menangkap citra simbol hanya 1-3 meter. Hal ini menyebabkan terjadi perubahan bentuk piksel pada citra yang berada di atas 3 meter sehingga menyebabkan sebagian citra simbol tidak dapat dikenali. Namun *error* ini dapat diatasi dengan cara mengganti kamera *web* yang memiliki resolusi lebih tinggi.

Dari data diatas terlihat bahwa *error* terjadi pada jarak 4 meter hingga 5 meter. Dan didapat *error* rata-ratanya adalah 22.22%

4.3. Pengujian Pengenalan Simbol Dengan Menggabung Citra Referensi di dalam Ruangan

Dari percobaan yang dilakukan didapatkan hasil dengan *error* rata-rata sebesar 12.844%. *Error* terjadi dikarenakan jarak efektif kamera menangkap citra simbol hanya 1-3 meter dan *keypoint* citra simbol lain yang seharusnya tidak digunakan ikut terhitung hal ini terjadi karena simbol yang dideteksi memiliki kesamaan karakteristik dengan simbol lainnya.

4.3.1. Pengujian Pengenalan Simbol di dalam Ruangan Waktu Pagi

Pada percobaan ini dilakukan dengan menggabungkan citra referensi semua simbol menjadi satu dan diujikan saat pagi hari pukul 08.00.

4.3.1.1. Pengujian Tidak Ada Simbol

Pengujian tidak ada simbol dilakukan untuk mengetahui ada tidaknya *keypoint* ketika didepan kamera tidak ada simbol, jika ada maka terdapat *error*.

Dari data diatas terlihat bahwa pengujian tidak ada simbol memiliki *error* sebesar 3.125 %.

4.3.1.2. Pengujian Ada Simbol

Pengujian ada simbol dilakukan dengan membandingkan 5 citra referensi simbol gabungan dengan tiap masing-masing simbol.

Dari data diatas terlihat bahwa *keypoint* referensi simbol gabungan jika dibandingkan dengan persimbol memiliki selisih yang sangat jauh. Apabila *keypoint* yang di dapat dibandingkan dengan *keypoint* per simbol akan memiliki akurasi *error* yang kecil namun cenderung nilai *keypoint*nya melebihi referensi, hal ini terjadi karena beberapa *keypoint* dari simbol lain terhitung. Jika di bandingkan dengan *keypoint* per simbol maka simbol pertama memiliki *error* rata-rata sebesar 5%, simbol kedua 20.83%, simbol ketiga 19.44%, simbol keempat 17.94%, simbol kelima 1.01%. Sehingga didapat *error* rata-rata sebesar 12.844%.

4.3.2. Pengujian Pengenalan Simbol di dalam Ruangan Waktu Malam

Pada percobaan ini dilakukan dengan menggabungkan citra referensi semua simbol menjadi satu dan diujikan saat malam hari pukul 20.00.

4.3.2.1. Pengujian Tidak Ada Simbol

Pengujian tidak ada simbol dilakukan untuk mengetahui ada tidaknya *keypoint* ketika didepan kamera tidak ada simbol, jika ada maka terdapat *error*.

Dari data diatas terlihat bahwa pengujian tidak ada simbol memiliki *error* sama besar dengan pada pagi hari yakni sebesar 3.125 %.

4.3.2.2. Pengujian Ada Simbol

Pengujian ada simbol dilakukan dengan membandingkan 5 citra referensi simbol dengan tiap masing-masing simbol

Dari data diatas terlihat bahwa *keypoint* referensi simbol gabungan jika dibandingkan dengan persimbol memiliki selisih yang sangat jauh. Apabila *keypoint* yang di dapat dibandingkan dengan *keypoint* per simbol akan memiliki akurasi *error* yang kecil namun cenderung nilai *keypoint*nya melebihi referensi, hal ini terjadi karena beberapa *keypoint* dari simbol lain terhitung. Jika di bandingkan dengan *keypoint* per simbol maka simbol pertama memiliki *error* rata-rata sebesar 5%, simbol kedua

20.83%, simbol ketiga 19.44%, simbol keempat 17.94%, simbol kelima 1.01%. Sehingga didapat *error* rata-rata sebesar 12.844%.

4.4. Pengujian Kecepatan Pengguna Menuju Toilet Ketika Menggunakan Sistem

Pada pengujian ini dilakukan oleh 5 pengguna, yang terdiri dari 3 pengguna yang sudah memakai sistem lebih dari 3 kali dengan 2 pengguna yang baru memakai sistem. Perhitungan waktu dilakukan dari sebelum pengguna mengenali simbol hingga pengguna mencapai toilet. Ilustrasi pengujian sistem dilakukan dengan 2 tahap yang pertama mencari dan mengenali simbol toilet dan kedua mencari simbol toilet pria atau wanita.

Dari data di atas dapat diketahui jika pengguna baru memiliki waktu lebih lama 2 kali lipat dari pengguna lama hal ini terjadi karena pengguna lama masih beradaptasi dengan sistem, proses adaptasi membutuhkan waktu cukup lama. Selain itu dalam mencari simbol pengguna baru juga beradaptasi dengan lingkungan, sehingga dalam mencari dan mengenali simbol ada yang berputar 360 derajat hingga simbol ditemukan.

4.5. Pengujian Jika Terdapat Gangguan Pada Simbol

Pengujian dilakukan dengan cara memberikan coretan pada simbol saat siang hari didalam ruangan hal ini dilakukan untuk mengukur keakuratan sistem ketika simbol mengalami gangguan berupa coretan. Pada pengujian ini menggunakan 2 simbol yang dicoret dengan 2 warna, yakni hitam dan merah.

Dari data diatas didapat jika terdapat gangguan pada simbol maka keypoint sistem akan turun hal ini terjadi karena sebagian karakteristik dari simbol tidak dapat dikenali dan didapat *error* rata-rata sebesar 34.02%.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil pengujian dan analisis tugas akhir ini dapat diambil beberapa kesimpulan yakni :

1. Proses ekstraksi fitur SURF pada suatu citra tidak terpengaruh cahaya.
2. Metode SURF membutuhkan waktu ekstraksi sebesar 2 hingga 3 detik.
3. Kemampuan efektif kamera menangkap citra pada sistem ini berjarak 1 hingga 3 meter.
4. Dari hasil pengujian pada sistem tugas akhir ini memiliki error sebesar 25.20% pada pengujian simbol dengan simbol referensinya sendiri dan error ditempat umum sebesar 22.22% sedangkan untuk pengujian citra referensi gabungan memiliki error yang paling rendah yakni 12.84%. Sehingga sistem ini memiliki error rata-rata 20% dan akurasi sebesar 80%

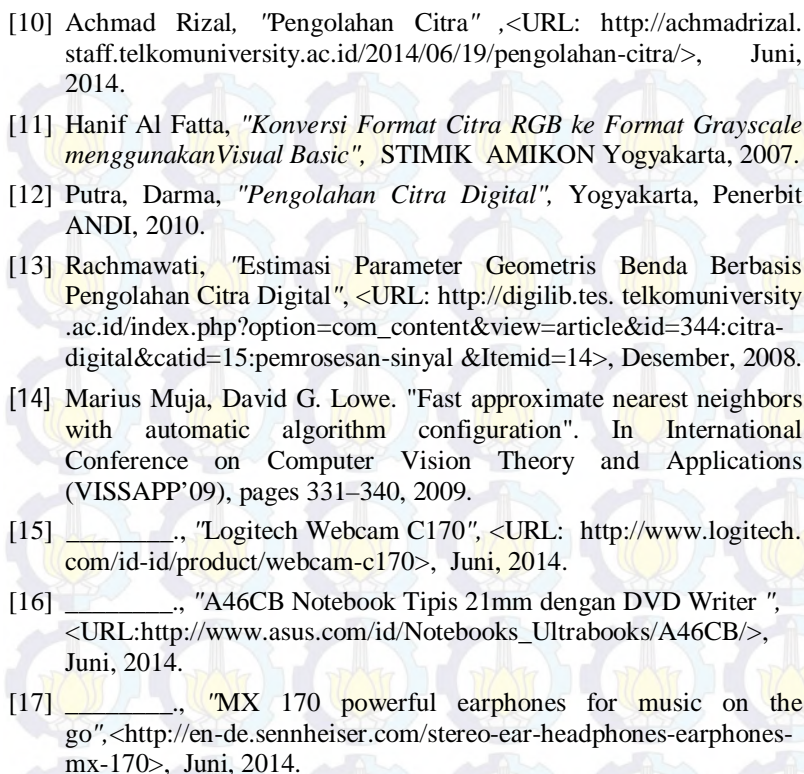
5.2. Saran

1. Karena metode yang digunakan terpengaruh dari kondisi cahaya disekitar, maka metode pengenalan simbol SURF yang bisa tidak terpengaruh oleh cahaya
2. Pengenalan simbol dengan metode SURF juga terpengaruh dengan hasil tangkapan citra, semakin jauh jarak tangkap kamera semakin kurang bagus citra yang didapatkan, maka disarankan metode pengenalan simbol ini tidak terpengaruh dengan kemampuan tangkap citra kamera.
3. Performa metode SURF yang tidak terjadi delay.
4. Mampu mengenali simbol sekaligus mampu mengetahui lokasi ruang toilet dengan mengenali arah panah yang terdapat pada simbol.
5. Untuk memperbaiki *error* disarankan menggunakan kamera dengan image sensor dan resolusi yang lebih tinggi.



DAFTAR PUSTAKA

- [1] Nigel H. Lovell, John W. Morley, Spencer C. Chen, Luke E. Hallum, and Gregg J. Suaning, "Biological–MachineSystems Integration: Engineeringthe NeuralInterface", Vol. 98, No. 3, March 2010.
- [2] H. W. Lee, S. B. Hong, D. W. Seo, W. S. Tae, S. C. Hong, "Mapping of functional organization in human visual cortex: Electrical cortical stimulation" *Neurology*, vol. 54, pp. 849–854, 2000.
- [3] Kulyukin, V., Kutiyawala, A., "Accessible Shopping Systems for Blind and Visually Impaired Individuals: Design Requirements and the State of the Art", *The Open Rehabilitation Journal* 3, 158–168, 2010.
- [4] H. Bay, T. Tuytelaars, L. Van Gool, "*SURF: Speeded up robust features*", In *ECCV*, pp. 404–417, 2006.
- [5] Vilar F. da Camara Neto, Mario Fernando M. Campos, "An Improved Methodology for Image Feature Matching", XXII Brazilian Symposium on Computer Graphics and Image Processing, 2009.
- [6] Yang Jian, Zhang David, Frangi Alejandro F, Yang Jingyu., "Two-Dimensional PCA: A new approach to appearancebased face representation and recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 131~137, 2004.
- [7] Wahyudi, Eko., Wirawan., Hendra Kusuma, "*Teknik Pengenalan Wajah Berbasis Local Binary Pattern*", Tugas Akhir ITS, Desember 2011.
- [8] Guillaume Heusch, Yann Rodriguez, S'ébastien Marcel, "Local Binary Pattern as an Image Preprocessing for Face Authentication", IDIAP Research Institute, Martigny, 2006.
- [9] Johnny Andrean Susanto, Kelvin Giovanni Lukman, "Digital Watermarking Untuk Melindungi Informasi Informasi Multimedia Dengan Metode Fast Fourier Transform (FFT)", <URL:http://eprints.mdp.ac.id/891/1/JURNAL%202009250032%20JOHNNY_ANDREAN_SUSANTO%20DAN%202009250027%20KELVIN GIOVANNI_LUKMAN.pdf>, 2003.

- 
- [10] Achmad Rizal, "Pengolahan Citra" ,<URL: <http://achmadrizal.staff.telkomuniversity.ac.id/2014/06/19/pengolahan-citra/>>, Juni, 2014.
- [11] Hanif Al Fatta, "*Konversi Format Citra RGB ke Format Grayscale menggunakan Visual Basic*", STIMIK AMIKON Yogyakarta, 2007.
- [12] Putra, Darma, "*Pengolahan Citra Digital*", Yogyakarta, Penerbit ANDI, 2010.
- [13] Rachmawati, "Estimasi Parameter Geometris Benda Berbasis Pengolahan Citra Digital", <URL: http://digilib.telkomuniversity.ac.id/index.php?option=com_content&view=article&id=344:citra-digital&catid=15:pemrosesan-sinyal&Itemid=14>, Desember, 2008.
- [14] Marius Muja, David G. Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration". In International Conference on Computer Vision Theory and Applications (VISSAPP'09), pages 331–340, 2009.
- [15] _____, "Logitech Webcam C170", <URL: <http://www.logitech.com/id-id/product/webcam-c170>>, Juni, 2014.
- [16] _____, "A46CB Notebook Tipis 21mm dengan DVD Writer ", <URL:http://www.asus.com/id/Notebooks_Ultrabooks/A46CB/>, Juni, 2014.
- [17] _____, "MX 170 powerful earphones for music on the go",<<http://en-de.sennheiser.com/stereo-ear-headphones-earphones-mx-170>>, Juni, 2014.

BIODATA PENULIS



Penulis dilahirkan di Surabaya pada tanggal 06 Oktober 1992. Sebagai anak ketiga dari tiga bersaudara, penulis mengawali pendidikan formal di SD Negeri Perak Barat 1, yang kemudian dilanjutkan di SMP Negeri 2 Surabaya, SMA Negeri 2 Surabaya dan pada tahun 2010 penulis diterima sebagai mahasiswa di jurusan Teknik Elektro ITS Surabaya. Selama kuliah penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar pada semester 5 hingga semester 8.

Email: tonny.feriandi@gmail.com

LAMPIRAN

```
#include "cv.h"
#include "highgui.h"
#include "ctype.h"
#include "stdio.h"
#include "stdlib.h"
#include "iostream"
#include "vector"
int waktu=0;
int toi=0;
int wan=0;
int pri=0;
int wc=0, men=0, women=0;
int j=0, k=0, l=0;
using namespace std;
static double dis=0;
CvFont font;
IplImage *image = 0, *frame = 0;

//==== compareSURF
//=====
double compareSURFDescriptors( const float* d1, const
float* d2, double best, int length )
{
    double total_cost = 0;
    assert( length % 4 == 0 );
    for( int i = 0; i < length; i += 4 )
    {
        double t0 = d1[i] - d2[i];
        double t1 = d1[i+1] - d2[i+1];
        double t2 = d1[i+2] - d2[i+2];
        double t3 = d1[i+3] - d2[i+3];
        total_cost += t0*t0 + t1*t1 + t2*t2 + t3*t3;
        // if( total_cost > best )
        //break;
    }
    return total_cost;
}
```

```

//==== NearestNeighbor
//=====
int naiveNearestNeighbor( const float* vec, int
laplaci an, const CvSeq* model_keypoints, const CvSeq*
model_descriptors )
{
    int length = (int)(model_descriptors-
>elem_size/si zeof(float));
    int i, neighbor = -1;
    double d, dist1 = 1e6, dist2 = 1e6;
    CvSeqReader reader, kreader;
    cvStartReadSeq( model_keypoints, &kreader, 0 );
    cvStartReadSeq( model_descriptors, &reader, 0 );

    for( i = 0; i < model_descriptors->total; i++ )
    {
        const CvSURFPoi nt* kp = (const
CvSURFPoi nt*)kreader.ptr;
        const float* mvec = (const float*)reader.ptr;
        CV_NEXT_SEQ_ELEM( kreader.seq->elem_si ze, kreader
);
        CV_NEXT_SEQ_ELEM( reader.seq->elem_si ze, reader
);
        if( laplaci an != kp->laplaci an )
            continue;
        d = compareSURFDescriptors( vec, mvec, di st2,
length );

        if( d < dist1 )
        {
            dist2 = dist1;
            dist1 = d;
            neighbor = i;
        }
        else if ( d < dist2 )
            dist2 = d;
    }
    di s=di s+di st1;
}

```

```

        if ( dist1 < 0.5*dist2 )
            return neighbor;
        return -1;
    }

    //==== find pairs
    //=====
    void findPairs( const CvSeq* objectKeypoints, const
CvSeq* objectDescriptors,
                    const CvSeq* imageKeypoints, const CvSeq*
imageDescriptors, vector<int>& ptpairs )
    {
        int i;
        CvSeqReader reader, kreader;
        cvStartReadSeq( objectKeypoints, &kreader );
        cvStartReadSeq( objectDescriptors, &reader );
        ptpairs.clear();

        for( i = 0; i < objectDescriptors->total; i++ )
        {
            const CvSURFPoint* kp = (const
CvSURFPoint*)kreader.ptr;
            const float* descriptor = (const
float*)reader.ptr;
            CV_NEXT_SEQ_ELEM( kreader.seq->elem_size,
kreader );
            CV_NEXT_SEQ_ELEM( reader.seq->elem_size, reader
);

            int nearest_neighbor = naiveNearestNeighbor(
descriptor, kp->laplacian, imageKeypoints,
imageDescriptors);
            if( nearest_neighbor >= 0 )
            {
                ptpairs.push_back(i);
                ptpairs.push_back(nearest_neighbor);
            }
        }
    }
}

```



```

int main()
{
    //cvNamedWindow( "video", CV_WINDOW_AUTOSIZE );
    CvCapture* capture;
    capture = cvCreateCameraCapture(1);
    CvSeq *objectKeypoints = 0, *objectDescriptors =
0;
    CvSeq *imageKeypoints = 0, *imageDescriptors = 0;
    CvMemStorage* storage = cvCreateMemStorage(0);
    int a,b,c,p,w,t;

while(1)
{
    frame = cvRetrieveFrame( capture );
    if(!frame )
    {
        IplImage* temp = cvLoadImage( "temp1.jpg", 1 );
        cvSaveImage("temp.jpg", temp);
        cvReleaseImage(&temp);
    }
    if( cvWaitKey( 1 ) == 'q' )
    {
        IplImage* temp = cvLoadImage( "temp1.jpg", 1 );
        cvSaveImage("temp.jpg", temp);
        cvReleaseImage(&temp);
        //break;
    }
    waktu++;
    if(waktu>5 && waktu < 15)
    {
        waktu=0;
        IplImage* img1 = cvCloneImage(frame);
        cvSaveImage("temp.jpg", img1);
        cvReleaseImage(&img1);
    }
    IplImage* object = cvLoadImage( "all.jpg",
CV_LOAD_IMAGE_GRAYSCALE );
    IplImage* object1 = cvLoadImage( "men.jpg",
CV_LOAD_IMAGE_GRAYSCALE );

```

```

IplImage* object2 = cvLoadImage( "women.jpg",
CV_LOAD_IMAGE_GRAYSCALE );
IplImage* tonny = cvLoadImage( "tonny.jpg",
CV_LOAD_IMAGE_GRAYSCALE );
IplImage* image = cvLoadImage( "temp.jpg",
CV_LOAD_IMAGE_GRAYSCALE );
IplImage* object_color =
cvCreateImage(cvGetSize(object), 8, 3);
IplImage* object_color1 =
cvCreateImage(cvGetSize(object1), 8, 3);
IplImage* object_color2 =
cvCreateImage(cvGetSize(object2), 8, 3);
IplImage* image_color =
cvCreateImage(cvGetSize(image), 8, 3);
cvCvtColor( object, object_color, CV_GRAY2BGR );
cvCvtColor( object1, object_color1, CV_GRAY2BGR );
cvCvtColor( object2, object_color2, CV_GRAY2BGR );
cvCvtColor( image, image_color, CV_GRAY2BGR );
CvSURFParams params = cvSURFParams(500, 1);
cvExtractSURF( object, 0, &objectKeypoints,
&objectDescriptors, storage, params );
cvExtractSURF( image, 0, &imageKeypoints,
&imageDescriptors, storage, params );

//=====
=====
//gambar pertama (merah) toilet
//=====
=====

vector<int> ptpairs;
findPairs( objectKeypoints, objectDescriptors,
imageKeypoints, imageDescriptors, ptpairs );
for( t = 0; t < (int)ptpairs.size(); t += 2 )
{
    CvSURFPoint* r1 = (CvSURFPoint*)cvGetSeqElem(
objectKeypoints, ptpairs[t] );
    CvSURFPoint* r2 = (CvSURFPoint*)cvGetSeqElem(
imageKeypoints, ptpairs[t+1] );
    CvPoint center;
    int radius;

```

```

        center.x = cvRound(r2->pt.x);
        center.y = cvRound(r2->pt.y);
        radius = cvRound(r1->size*1.2/9.*2);
        cvCircle( image_color, center, radius,
cvScalar(0,0,255,0), 1, 8, 0 );
        toi=center.x;
    }

    for( a = 0; a < objectKeypoints->total; a++ )
    {
        CvSURFPoint* r = (CvSURFPoint*)cvGetSeqElem(
objectKeypoints, a );
        CvPoint center;
        int radius;
        center.x = cvRound(r->pt.x);
        center.y = cvRound(r->pt.y);
        radius = cvRound(r->size*1.2/9.*2);
        cvCircle( object_color, center, radius,
cvScalar(0,0,255,0), 1, 8, 0 );
    }

    //=====
    //gambar kedua (hijau) pria
    //=====

    cvExtractSURF( object1, 0, &objectKeypoints,
&objectDescriptors, storage, params );
    findPairs( objectKeypoints, objectDescriptors,
imageKeypoints, imageDescriptors, ptpairs );

    for( p = 0; p < (int)ptpairs.size(); p += 2 )
    {
        CvSURFPoint* r11 = (CvSURFPoint*)cvGetSeqElem(
objectKeypoints, ptpairs[p] );
        CvSURFPoint* r21 = (CvSURFPoint*)cvGetSeqElem(
imageKeypoints, ptpairs[p+1] );
        CvPoint center1;
        int radius1;
    }

```

```

        center1.x = cvRound(r21->pt.x);
        center1.y = cvRound(r21->pt.y);
        radius1 = cvRound(r11->size*1.2/9.*2);
        cvCircle( image_color, center1, radius1,
cvScalar(0,255,0,0), 1, 8, 0 );
        pri=center1.x;
    }

    for( b = 0; b < objectKeypoints->total; b++ )
    {
        CvSURFPoi nt* r1 = (CvSURFPoi nt*)cvGetSeqElem(
objectKeypoints, b );
        CvPoi nt center;
        int radius;
        center.x = cvRound(r1->pt.x);
        center.y = cvRound(r1->pt.y);
        radius = cvRound(r1->size*1.2/9.*2);
        cvCircle( object_color1, center, radius,
cvScalar(0,255,0,0), 1, 8, 0 );
    }

    //=====
    //gambar ketiga (biru) wanita
    //=====

    cvExtractSURF( object2, 0, &objectKeypoints,
&objectDescriptors, storage, params );
    findPairs( objectKeypoints, objectDescriptors,
imageKeypoints, imageDescriptors, ptpairs );
    for( w = 0; w < (int)ptpairs.size(); w += 2 )
    {
        CvSURFPoi nt* r11 = (CvSURFPoi nt*)cvGetSeqElem(
objectKeypoints, ptpairs[w] );
        CvSURFPoi nt* r21 = (CvSURFPoi nt*)cvGetSeqElem(
imageKeypoints, ptpairs[w+1] );
        CvPoi nt center1;
        int radius1;

```



```

        center1.x = cvRound(r21->pt.x);
        center1.y = cvRound(r21->pt.y);
        radius1 = cvRound(r11->size*1.2/9.*2);
        cvCircle( image_color, center1, radius1,
cvScalar(255,0,0,0), 1, 8, 0 );
        wan=center1.x;
    }

    for( c = 0; c < objectKeypoints->total; c++ )
    {
        CvSURFPoint* r1 = (CvSURFPoint*)cvGetSeqElem(
objectKeypoints, c );
        CvPoint center;
        int radius;
        center.x = cvRound(r1->pt.x);
        center.y = cvRound(r1->pt.y);
        radius = cvRound(r1->size*1.2/9.*2);
        cvCircle( object_color2, center, radius,
cvScalar(255,0,0,0), 1, 8, 0 );
    }

    if(a>64)a=64;
    if(p>b)p=b;
    if(w>c)w=c;
    int derajattoi=(toi-frame->width/2)/(3.5555*2);
    int derajattpri=(pri-frame->width/2)/(3.5555*2);
    int derajattwan=(wan-frame->width/2)/(3.5555*2);
    char
text1[80],text2[80],text3[80],text4[80],text5[80],text6[
80];
    cvInitFont(&font,CV_FONT_HERSHEY_SIMPLEX,
0.6,0.6,0,2);
    sprintf(text1,"referensi toilet : %d, toilet : %d
",a,t);
    sprintf(text2,"referensi pria : %d, pria : %d
",b,p);
    sprintf(text3,"referensi wanita : %d, wanita : %d
",c,w);
    sprintf(text4,"lokasi toilet: %d ",derajattoi);
    sprintf(text5,"lokasi pria : %d ",derajatpri);
    sprintf(text6,"lokasi wanita: %d ",derajatwan);

```

```

        cvPutText (i mage_col or, text1, cvPoi nt(10, 20),
&font, cvSca lar(0,0,0));
        cvPutText (i mage_col or, text2, cvPoi nt(10, 40),
&font, cvSca lar(0,0,0));
        cvPutText (i mage_col or, text3, cvPoi nt(10, 60),
&font, cvSca lar(0,0,0));
        cvPutText (i mage_col or, text4, cvPoi nt(10, 80),
&font, cvSca lar(0,0,0));
        cvPutText (i mage_col or, text5, cvPoi nt(10, 100),
&font, cvSca lar(0,0,0));
        cvPutText (i mage_col or, text6, cvPoi nt(10, 120),
&font, cvSca lar(0,0,0));
        Ipl Image* gabung = cvCreateI mage(
cvSi ze(i mage_col or->wi dth+tonny->wi dth, i mage_col or-
>hei ght), 8, 3 );
        for(i nt x=0; x<i mage_col or->wi dth; x++){
            for(i nt y=0; y<i mage_col or-
>hei ght; y++)
            {
                gabung-
>i mageData[gabung->wi dthStep*y+x*gabung-
>nChannel s+1]=i mage_col or->i mageData[i mage_col or-
>wi dthStep*y+x*i mage_col or->nChannel s+1];
                gabung-
>i mageData[gabung->wi dthStep*y+x*gabung-
>nChannel s+2]=i mage_col or->i mageData[i mage_col or-
>wi dthStep*y+x*i mage_col or->nChannel s+2];
                gabung-
>i mageData[gabung->wi dthStep*y+x*gabung-
>nChannel s+3]=i mage_col or->i mageData[i mage_col or-
>wi dthStep*y+x*i mage_col or->nChannel s+3];
            }
        }
        for(i nt x=0; x<tonny->wi dth; x++){
            for(i nt y=0; y<tonny-
>hei ght; y++)
            {
                gabung-
>i mageData[gabung->wi dthStep*y+(x+frame->wi dth)*gabung-

```

```

>nChannel s+1]=tonny->i mageData[tonny-
>wi dthStep*y+x*tonny->nChannel s+1];
                                gabung-
>i mageData[gabung->wi dthStep*y+(x+frame->wi dth)*gabung-
>nChannel s+2]=tonny->i mageData[tonny-
>wi dthStep*y+x*tonny->nChannel s+2];
                                gabung-
>i mageData[gabung->wi dthStep*y+(x+frame->wi dth)*gabung-
>nChannel s+3]=tonny->i mageData[tonny-
>wi dthStep*y+x*tonny->nChannel s+3];
                                }
                                }
    for (i nt x=0; x<obj ect_col or->wi dth; x++){
                                for (i nt y=0; y<obj ect_col or-
>hei ght; y++)
                                {
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny->hei ght)+(x+frame-
>wi dth)*gabung->nChannel s+1]=obj ect_col or-
>i mageData[obj ect_col or->wi dthStep*y+x*obj ect_col or-
>nChannel s+1];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny->hei ght)+(x+frame-
>wi dth)*gabung->nChannel s+2]=obj ect_col or-
>i mageData[obj ect_col or->wi dthStep*y+x*obj ect_col or-
>nChannel s+2];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny->hei ght)+(x+frame-
>wi dth)*gabung->nChannel s+3]=obj ect_col or-
>i mageData[obj ect_col or->wi dthStep*y+x*obj ect_col or-
>nChannel s+3];
                                }
                                }
    for (i nt x=0; x<obj ect_col or1->wi dth; x++){
                                for (i nt y=0; y<obj ect_col or1-
>hei ght; y++)
                                {
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame->wi dth)*gabung-

```

```

>nChannel s+1]=obj ect_col or1->i mageData[obj ect_col or1-
>wi dthStep*y+x*obj ect_col or1->nChannel s+1];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame->wi dth)*gabung-
>nChannel s+2]=obj ect_col or1->i mageData[obj ect_col or1-
>wi dthStep*y+x*obj ect_col or1->nChannel s+2];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame->wi dth)*gabung-
>nChannel s+3]=obj ect_col or1->i mageData[obj ect_col or1-
>wi dthStep*y+x*obj ect_col or1->nChannel s+3];
                                }
                                }
    for(i nt x=0; x<obj ect_col or2->wi dth; x++){
        for(i nt y=0; y<obj ect_col or2-
>hei ght; y++)
            {
                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame-
>wi dth+obj ect_col or1->wi dth)*gabung-
>nChannel s+1]=obj ect_col or2->i mageData[obj ect_col or2-
>wi dthStep*y+x*obj ect_col or2->nChannel s+1];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame-
>wi dth+obj ect_col or1->wi dth)*gabung-
>nChannel s+2]=obj ect_col or2->i mageData[obj ect_col or2-
>wi dthStep*y+x*obj ect_col or2->nChannel s+2];
                                gabung-
>i mageData[gabung->wi dthStep*(y+tonny-
>hei ght+obj ect_col or->hei ght)+(x+frame-
>wi dth+obj ect_col or1->wi dth)*gabung-
>nChannel s+3]=obj ect_col or2->i mageData[obj ect_col or2-
>wi dthStep*y+x*obj ect_col or2->nChannel s+3];
                                }
                                }
//=====Perintah Suara==

```



```

//=====toi let==
=====
    if(derajattoi > -15 && derajattoi < 15 && t > 14 && t < 35
    && wc == 0) {PI aySound(TEXT("t.wav"),
    NULL, SND_SYNC); wc=1; j=derajattoi; men=1; women=1; }
    if(derajattoi > 15 && derajattoi < 45 && t > 14 && t < 35
    && wc == 0) {PI aySound(TEXT("tk1.wav"),
    NULL, SND_SYNC); wc=1; j=derajattoi; men=1; women=1; }
    if(derajattoi > -45 && derajattoi < -15 && t > 14 &&
    t < 35 && wc == 0) {PI aySound(TEXT("tk-1.wav"),
    NULL, SND_SYNC); wc=1; j=derajattoi; men=1; women=1; }
    if(t < 12 || abs(j - derajattoi) > 5) {wc=0; }
//=====pria=====
=====
    if(derajatpri > -15 && derajatpri < 15 && p > 9 &&
    men == 0) {PI aySound(TEXT("p.wav"),
    NULL, SND_SYNC); men=1; k=derajatpri; wc=1; }
    if(derajatpri > 15 && derajatpri < 45 && p > 9 &&
    men == 0) {PI aySound(TEXT("pk1.wav"),
    NULL, SND_SYNC); men=1; k=derajatpri; wc=1; }
    if(derajatpri > -45 && derajatpri < -15 && p > 9 &&
    men == 0) {PI aySound(TEXT("pk-1.wav"),
    NULL, SND_SYNC); men=1; k=derajatpri; wc=1; }
    if(p < 9 || abs(k - derajatpri) > 5) {men=0; }
//=====wani ta=====
=====
    if(derajatwan > -15 && derajatwan < 15 && w > 6 &&
    women == 0) {PI aySound(TEXT("w.wav"),
    NULL, SND_SYNC); women=1; l=derajatwan; wc=1; }
    if(derajatwan > 15 && derajatwan < 45 && w > 6 &&
    women == 0) {PI aySound(TEXT("wk1.wav"),
    NULL, SND_SYNC); women=1; l=derajatwan; wc=1; }
    if(derajatwan > -45 && derajatwan < -15 && w > 6 &&
    women == 0) {PI aySound(TEXT("wk-1.wav"),
    NULL, SND_SYNC); women=1; l=derajatwan; wc=1; }
    if(w < 6 || abs(l - derajatwan) > 5) {women=0; }
//=====
    system("cls");
    printf("toi let      : %d \n", wc);
    printf("pria : %d \n", men);

```

```

printf("wani ta      : %d \n",women);
cvShowImage( "gabung", gabung);
//cvShowImage( "gabunga", frame);
cvReleaseImage( &tonny );
cvReleaseImage( &object );
cvReleaseImage( &object1 );
cvReleaseImage( &object2 );
cvReleaseImage( &gabung);
cvReleaseImage( &image );
cvReleaseImage( &image_color );
cvReleaseImage( &object_color );
cvReleaseImage( &object_color1 );
cvReleaseImage( &object_color2 );
}
cvReleaseCapture( &capture );
cvDestroyWindow( "video" );
}

```

